Virtual Treeview

Virtual Treeview

Table of Contents

Virtual Treeview

1 Introduction

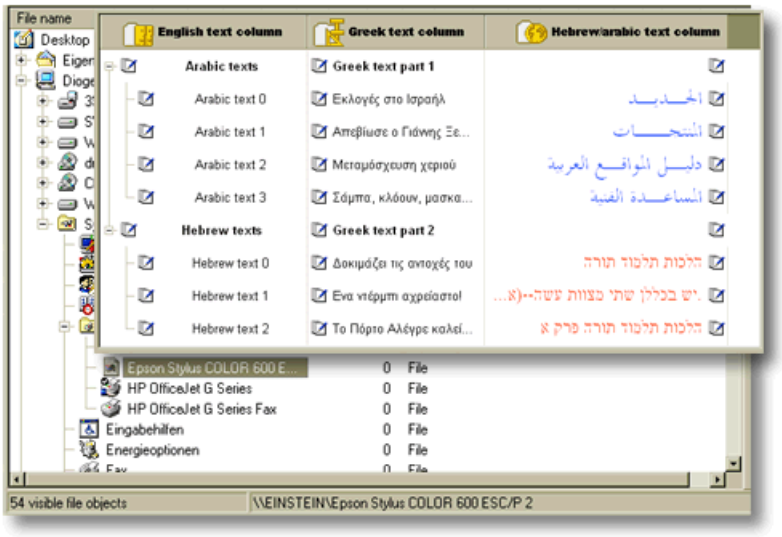
Virtual Treeview is a tree view control built from ground up.

Description

More than 3 years of development made it one of the most flexible and advanced tree controls available today. Virtual Treeview starts off with the claim to improve many aspects of existing solutions and introduces some new technologies and priniciples which were not available before.

As the name already indicates, this control uses a different paradigm for tree management than other controls of this kind. It does not know anything about the data it manages (except its size), not even the captions of a node. Everything is retrieved from the application via events (or descendants via overridden methods).

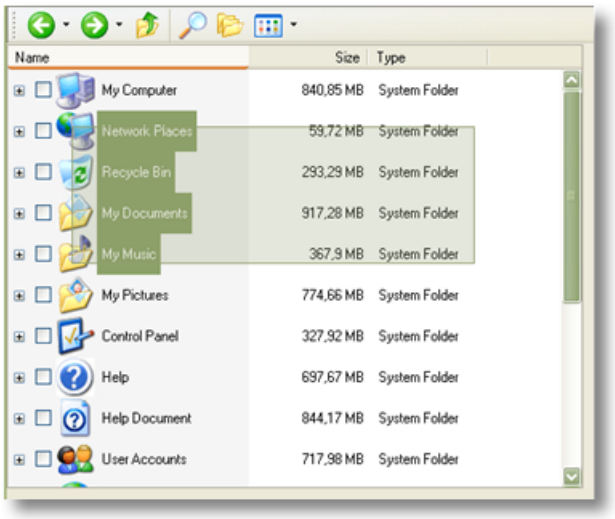
Virtual Treeview has been carefully designed and thoroughly tested. The control proved its concept as well as everyday fitness already in many commercial products and freeware projects.



Virtual Treeview can be characterized by the following core capabilities:

- Extremely fast and designed for high speed access.
- Memory sparing which is the premise for speed and capacity.
- A high capacity control.
- Highly customizable.
- Designed for professionals, implements a virtual paradigm with a new serialization concept.

- Newest technologies and platforms are supported (e.g. Windows XP).
- Unique features like Unicode, right-to-left directionality and layout, alpha blending and OLE drag'n drop and clipboard operations.



Homepage: www.soft-gems.net

E-Mail: support@soft-gems.net

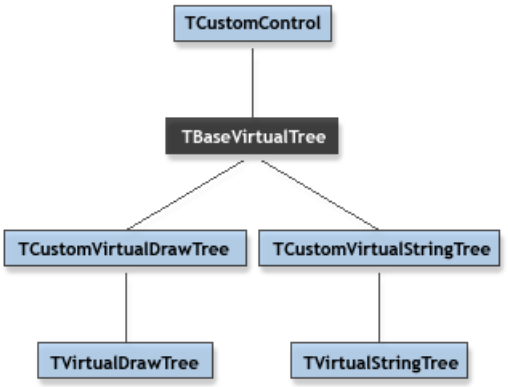
Support center: support.soft-gems.net

News group: delphi-gems.support.virtualtreeview

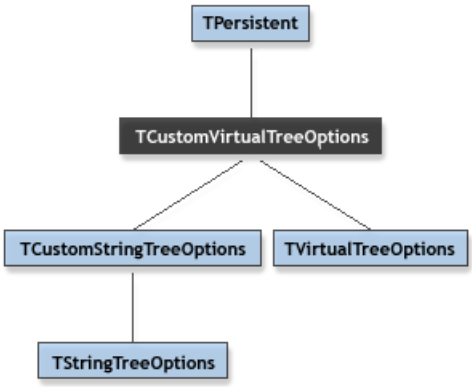
Web based forums: support.soft-gems.net/forums

Issue Tracker: support.soft-gems.net/mantis

2 Features overview



Virtual Treeview base hierarchy



Virtual Tree Options hierarchy

Virtual Treeview is "pure VCL" which means it is not based on any of the system controls but was written from scratch.

Description

As the name already indicates, this control uses a different paradigm for tree management than other controls of this kind. It does not know anything about the data it manages (except its size), not even the captions of a node. Everything is retrieved from the application via events (or descendants via overridden methods). Virtual Treeview has been carefully designed and thoroughly tested. The control proved its concept as well as everyday fitness already in many commercial products and freeware projects.The following list summarizes in categories the most important features:

General

- Virtual Treeview is extremely fast. Adding one million nodes takes only 700 milliseconds* ! This makes it currently the fastest treeview publicly available on the Delphi/BCB market.
- Virtual Treeview has a very small memory foot print. By only allocating about 60 bytes per node (in the string tree, the base tree uses only 56 bytes) it is well prepared to hold a million of them.
- Virtual Treeview is optimized for high speed access. It takes as few as 0.5 seconds to traverse one million nodes* depending on needed validation and node validation states.
- Multiselection is supported, including constrained selection so that only nodes of a certain initial level can be selected. A lot of effort has been put into the development of effective algorithms e.g. to allow for modifying an already large selection set still interactively.
- Drawing the entire tree to a bitmap or the printer is supported by the central PaintTree method. The messages WM_PRINT and WM_PRINTCLIENT are handled correctly which allow things like drawing a tree into a bitmap (e.g. for layered windows or to implement animated drop down of controls which use VT as drop down control).
- There is an OnHint event to display node specific hints.
- There is an OnGetHelpContext event to retrieve node specific help context IDs. This includes automatic tree and window parent control traversal as is invoked when the user pressed F1
- There is an OnGetPopupMenu event to retrieve node specific popup menus, includes automatic tree traversal.
- Middle and right mouse buttons can be used in addition to the left button and support everything which is possible

with the left button (dragging, selection etc.). These alternative buttons can be switched, of course.

- A fixed background image can be used in the tree and can be given a certain offset, e.g. to simulate shared backgrounds.
- Hot style for nodes is supported (just like links in a browser window). A special cursor can be assigned for this task.
- String trees support so called static text which appears after a node's caption (in every column) and which can be formatted differently to the caption but cannot be edited, selected etc.
- An auto span column mode is supported which allows a column to take up more space for its caption if there are empty columns to its right. This avoids clipping of long captions but still allows using multiple columns.
- A node can be selected in every column (this is switchable) as well as edited, making Virtual Treeview some kind of a grid too. The tabulator key can be used to switch the focus between cells. A special option (toGridExtensions) exist to support grid specific tasks.
- Nodes can have individual heights and the vertical alignment of a node's images and lines can be adjusted individually.
- Virtual Treeview exposes its internal states like pending drag or edit events, multi selection or expanding in progress. Using this information an application can optimize its code execution (state updates etc.).
- Sorting a node is supported via an application defined compare call back. Additionally, a tree can be set to auto sort.
- Hints can contain multiple lines of text and mirror the alignment and directionality of the node or column they are displayed for. For their animation sliding and alpha blending is available.
- Incremental search with various options and directions is available too.
- Auto scrolling of the client area happens when the mouse is near the borders while dragging and draw selecting (multi selection).
- Default node height and default node text for string trees can be used to avoid setting many nodes explicitly to the same start value.

Newest technologies

- For smooth animations (e.g. hint fading) Virtual Treeview uses hand optimized MMX assembler routines. This code is also used to draw the translucent selection rectangle in multi selection mode. This is very much like what Windows 2000 and Windows XP support but works also on Windows 95/98/Me.
- An alpha blended image of the tree window is shown while doing drag and drop. On Windows 2000 and Windows XP IDropTargetHelper(see IDropTargetHelper Interface, page 702) and IDragSourceHelper(see IDragSourceHelper Interface, page 701) interfaces are supported which allow for some very neat effects (as used by Explorer). On older consumer Windows versions the drag image is simulated by the tree but underlies there some minor limitations.
- Virtual Treeview supports Windows XP themes. It acts properly on theme changes and uses for all visual elements which are themed the correct image by using native APIs. Under other Windows systems these styles are supported by separate legacy code. Theme awareness can be switched.

Unicode

- TVirtualStringTree(see TVirtualStringTree Class, page 402) is implemented using Unicode/wide strings exclusively.
- The tree saves and reads all Unicode properties (e.g. column captions, default node text and the like) correctly to/from DFM.
- All Unicode drawing fully supports bidirectionality (i.e. right-to-left drawing), column alignment (left, center, right) and correctly aligned hints. Of course also this feature is available on Windows 95/98/Me.

- On Windows NT/2000/XP multiline captions are fully supported (on Win9x/Me there is limited support).
- In order to have also Unicode editing capabilities Virtual Treeview supports the TNT controls written by Troy Wolbrink. This support can be enabled via a compiler switch named TntSupport. You must download and install the TNT package first, however.

Drag'n drop and clipboard support

- OLE drag and drop and OLE clipboard transfers are supported with the tree as source and target. Alternatively, VCL drag'n drop can still be used for compatibility.

These formats are support by the standard implementation:

- Native serialized format (CF_VIRTUALTREE(see CF_VIRTUALTREE Variable, page 658) and CF_VTREFERENCE(see CF_VTREFERENCE Variable, page 659)), which is a compact form to exchange data between Virtual Treeviews (also between applications). Two storage formats are available: HGlobal and IStream.
- Plain ANSI text string format.
- Plain Unicode text string format.
- Rich Text (RTF) string format (with Unicode text).
- HTML text string format (UTF-8). This is the preferred clipboard format for Word 2000 etc. and allows copy and paste tree content to a word document with nearly no application code.

There is a registration scheme which allows descendants to specify and implement their own clipboard formats. Via a drop handler the application can accept any OLE format without deriving an own tree class. In order to aid processing of the native tree data specialized methods are implemented. See also: ProcessOLEData and ProcessDrop.Dropmarks show during drag'n drop where data will be inserted. This works also with VCL drag'n drop. The drop target model has been extended to allow drop actions above, below or on a node. Meanwhile vendors of other treeview controls have started using this little but powerfull idea too.Auto expand of nodes which are the drop target for more than an adjustable time interval is performed if enabled.

Header and columns

- Multiple columns are supported by an own header implementation. This header takes up space in the non-client area of the tree control and supports various buttons styles (standard listview thick buttons, flat buttons, plates, Windows XP style and owner draw).
- Columns can appear in every order in the tree window.
- Each column can be hidden including the main column which holds the actual tree.
- Each column can become the main column.
- Columns can be shown also without the header.
- Columns can have various options (visible, clickable, resizable, draggable etc.).
- You can set individual alignments for each column as well as right-to-left or left-to-right directionality (again: available also on non-middle-east and older Windows consumer systems).
- Each column can have an own color.
- The header as well as the columns collection class and the actual column classes support streaming. This is independant from the treeview streaming.

- Each column can individually be customized by the application. An advance custom draw handling is implemented, which allows for very sophisticated effects, including animations.

Check support

- Each node in the tree can have its own check type. This can either be check box (also tristate), radio button or node button. These types can freely be mixed so you can for instance have a node with 10 nodes of which 5 comprise a radio group (where only one of these 5 nodes can be checked) and the other 5 nodes can have a check box (or no check type at all).
- Mixed (tri-state) check boxes with proper handling for partial checking of child nodes are supported (as often used in install and backup programs).
- Automatic state change propagation for mixed check button type is possible (if enabled).
- Check events OnChecking and OnCheck events are supplied too.
- For special purposes a small flat button can be used, which is called a node button.
- 7 different kinds of check images are possible. Dark and light check marks, dark and light tick marks, flat check images, Windows XP style check images and application defined check images. For an overview see property CheckImageKind.

Design time

- Virtual Treeview's properties and methods are registered with Delphi categories (Delphi 5 and BCB 5 or higher).
- A special property editor for the clipboard formats is included which allows a simple format choice. This is particularly important since the available clipboard formats must be given as strings and it is also quite handy to have a list of available formats, even if they are not enabled yet (to know what can be enabled).

Customization

- Custom draw and paint cycles are supported via paint events (for the entire tree and for each node)
- Apart from the built-in check types a user defined check image can be used which is supported by a separate image list.
- Each button in the header can be drawn individually.
- Three different lines styles are available: dotted lines, solid lines and application defined lines.
- Applications and descendants can provide their own node editor (which has not necessarily to be a single control) by handling the OnCreateEditor event or overriding DoCreateEditor. This allows to completely replace node editing by own (business) rules.

*

- Applications and descendants can provide their own drag manager interface by handling the
- OnCreateDragManager event or overriding DoCreateDragManager. This allows to customize the entire OLE drag handling of the

- tree. Note: VCL drag'n drop is managed by the VCL so this cannot be customized.
- Applications and descendants can provide their own data object interface by handling the OnCreateDataObject event or overriding DoCreateDataObject. This allows to provide own clipboard formats.
- There is registration function (RegisterVTClipboardFormat(see RegisterVTClipboardFormat Function, page 568)) which allows to register tree descendants with own clipboard and/or storage formats. Applications get provide own clipboard formats (without deriving new tree classes) by handling the GetUserClipboardFormats event.
- Applications and descendants can completely modify the tree's key handling by handling the OnKeyAction event or overriding DoKeyAction. This works also for incremental search.
- Applications and descendants can customize the tree's background which is not covered by nodes by handling the OnPaintBackground event or overriding DoPaintBackground. For nodes there are further events for customization.
- Applications and descendants can customize how the string tree shortens too long captions by handling the OnShortenString event or overriding DoShortenString.

Scrolling

- Flat scroll bars are supported. But since they conflict with Windows XP this support is switched off by a compiler symbol (UseFlatScrollbars). Enable this symbol if you really want to use flat scroll bars before compiling the tree unit.
- Every scroll operation triggers an OnScroll event. This allows to synchronize trees with other controls.
- There are properties (e.g. OffsetXY) which allow to scroll the tree content to any postion in code without sending messages around.

Streaming

- Sophisticated tree content serialization has been implemented to allow saving and restoring a tree to/from streams. This includes also user data as long as it can be written to a stream.
- Virtual Treeview allows also to add data from stream instead replacing the entire content.
- The internal format of the stream is chunk based which makes it very flexible for future enhancements but still keeps compatibility with older implementations.
- There is a user chunk which takes data written to the stream in the OnSaveNode event. The data of this user chunk is can be read in OnLoadNode.

Developer support

- Special care has been taken to format the source code of Virtual Treeview consistently.
- A large part of the entire implementation are comments which describe the inner workings.
- Methods and properties are consequently ordered alphabetically within their scope (private, protected, public, published). The only exception are the constructors and destructors which always appear at the top of the public section in the class declaration and are always the first methods in the class implementation.
- For every event there is a virtual method which calls the event handler. This allows descendants to get notice of every event without assigning a handler. The names of these methods correspond directly to the events by using the pattern: Do*EventName*.

- Many measures have been taken to ensure Borland C++ Builder compatibility. This is particularly difficult because the automatic translation from Delphi to C++ code in BCB is buggy.
- There is an easy and powerfull mechanism for descendants writers to allocate their own data on a per node basis. Simply call AllocateInternalData to register your needs. This will not influence existing or future application code if it consequently uses GetNodeData for user data access.

Editing

- Application defined editors are supported via an edit link interface. A generic (non-Unicode) editor implementation is available too.
- Every column in the tree is editable if enabled (see toExtendedFocus).
- By supporting the TNT controls library (see chapter Unicode above) it is also possible to have full Unicode editing capabilities.

Utilities

For your convenience some of the internally used functions which are of general interest are exposed.

- AlphaBlend(see AlphaBlend Function, page 565): a general purpose procedure to blend a source onto a target bitmap using several different modes.
- DrawTextW(see DrawTextW Function, page 566): a partial implementation of the DrawText API which supports Unicode. This method is not used on Windows NT/2000/XP machines.
- ShortenString(see ShortenString Function, page 568): a general purpose function which makes a given WideString fitting into a given space. This is partially implemented by the Windows DrawText API but takes additionally care for right-to-left alignment and works with Unicode also on Windows 95/98/Me.

* Times given here are taken on a Windws XP professional system running on an Athlon 650 MHz with 256MB RAM. All possible optimization were applied.

3 Installation

Virtual Treeview is designed for Delphi 4 and higher and can also be used with Borland C++ Builder 4 and up.

Description

It is however not designed to work directly with Kylix or Delphi for .NET. You will have to use a special descendant written by Dmitri Dimitrienko for Kylix support. Currently there is no .NET version available.

The initial core source files are:

Compilers.inc

Include file which contains various compiler switches which determine the target compiler and the target operating system.

VTConfig.inc

Include file which contains version neutral compiler switches which control certain things that can be compiled into the tree view (e.g. Windows XP theme support, Unicode controls, a specialized node memory manager etc.).

StrEditD4.dfm

Form file for the Delphi 4 TStrings property editor.

StrEditD4.pas

Delphi 4 TStrings property editor.

VirtualTrees.dcr

Component image for the tree components.

VirtualTrees.pas

The actual implementation of Virtual Treeview and its descendants and support classes.

VirtualTrees.res

Resource file containing some check and miscellaneous images used for all Virtual Treeviews.

VirtualTreesD4.*

Run time package for Delphi 4.

VirtualTreesD4D.*

Design time package for Delphi 4.

VirtualTreesD5.*

Run time package for Delphi 5.

VirtualTreesD5D.*

Design time package for Delphi 5.

... similar for all other Delphi versions except Delphi 8. Package files for Delphi 2005 are using number 9 as version identifier. For Borland C++ Builder there are similar files (e.g. VirtualTreesC4.bpk, VirtualTreesC4.cpp, VirtualTreesC4.res).

VirtualTreesReg.pas

Registration unit for some property editors and categories.

 VTHeaderPopup.pas

Unit containing a TPopupMenu descendant which provides a convenient way to implement a header popup used to switch visibility of columns.

Installation

The main Virtual Treeview distribution comes with an installation program and installs the components automatically into the selected and available target IDEs.

4 Version history

Version 4.3.0 - 4.4.2 (December 2004 - November 2005)

- Improvement: fixed column implementation completed (code donation by Igor Savkic)
- Change: OnMouseWheel published.
- Improvement: Painting of normal, selected, state and overlay image is now done using standard image list access.

Description

This allows to use specialized image lists (e.g. with full alpha channel support).

- Improvement: ShowScrollbar calls with conditional defines extracted into a new method. Added event that can be hooked by the application to get notified if a scrollbar is about to show or hide. Introduced OnShowScrollbar event.
- Improvement: OnGetImageEx event to allow specifying a custom imagelist.
- Improvement: GetFirstChecked, GetNextChecked, ClearChecked helper methods (code donation by Azza).
- Bug fix: Reselection of a node in multi selection node did not refresh its visual selection appearance.
- Bug fix: root node total count not updated during load of streamed nodes.
- Bug fix: When loading a node from stream the initial total height is always set to the current default height of the tree, not the height of the node that is being loaded.
- Bug fix: Mantis #260, TBaseVirtualTree.ReadChunk(see TBaseVirtualTree.ReadChunk Method , page 229) has applied total height of loaded nodes multiple times.
- Change: Moved DoCancelEdit and DoEndEdit to the protected section. Don't know it ever could end up in the public section. Use CancelEditNode and EndEditNode instead.
- Improvement: Hint window class dynamically assignable. TBaseVirtualTree.GetHintWindowClass(see TBaseVirtualTree.GetHintWindowClass Method , page 206)
- Change: A few GetPrevious* methods were still testing for an initialized parameter.
- Change: OnMouseWheel published.
- Improvement: Painting of normal, selected, state and overlay image is now done using standard image list access. This allows to use specialized image lists (e.g. with full alpha channel support).

Version 4.0.16 - 4.3.0 (December 2003 - December 2004)

- Improvement: Delphi 2005 compatibility.
- Bug fix: InternalData may return nil, so its result must be checked before accessing it.
- Bug fix: WM_CURSOR in TVTHeader.HandleMessage(see TVTHeader.HandleMessage Method , page 553) used the screen standard cursors as default instead that of the tree.
- Change: If the hot tracking cursor is crDefault when hot tracking is enabled then that of the tree is used instead.
- Bug fix: TVirtualTreeColumn.SetIndex removed as it caused reindexing of the position array (which is wrong).
- Bug fix: check for existing window handle before posting a message for the node editor.
- Change: published events OnAdvancedHeaderDraw and OnHeaderDrawQueryElements in TVirtualDrawTree(see TVirtualDrawTree Class, page 323).
- Improvement: tree state tsCheckPropagation is now only reset after a tristate check operation has finished (before the final OnChecked event). Therefore the tree state will include tsCheckPropagation while child nodes are checked or unchecked.
- Change: ExecuteAction fixed (incorrect conditional definitions)
- Change: DoBeforeItemErase was in the wrong place.

- Bug fix in InternalDisconnectNode: When an invisble node is removed from its parent the height of this parent node no longer is changed.
- Bug fix: Char handling for incremental search killed the dead char due to a problem with ToASCII.
- Improvement: Removed ParentBackground property (also for D7), it is useless because of the own background handling of VT
- Improvement: (better multimonitor support) Checks for true screen location for the hint.
- Bug fix: TVirtualTreeColumn.SetOptions + : Check for a valid window handle of the tree before doing invalidation.
- Improvement: In VT.WMKeyDown additional checks for page up/down, to scroll not more than what fits in one page under all conditions.
- Bug fix: In VT.HandleMouseDown check for assigned hit node before doing selection with alt key.
- Bug fix: VST.DoNewText, inserted call to UpdateHorizontalScrollbar to account for edited nodes, which now have a significant other length.
- Change: Moved some methods to higher visibility.
- Improvement: non-tiled background images (code donation by Richard Pringle).
- Improvement: Configuration compiler switches are now located in an additional file (VTConfig.inc).
- Improvement: Reset of all global objects to nil on finalization. Explicit initialization of Initialized(see Initialized Variable, page 662) and NeedToUnitialize(see NeedToUnitialize Variable, page 664) because of trouble when VT is used in dynamically loaded packages.
- Bug fix: Dragging did not work with full row selection and toFullRowDrag switched on while drag mode is dmManual.
- Improvement: Mouse button flags are now passed through OnDragOver and OnDragDrop.
- Bug fix: The internal node edit now uses clWindowText instead of clBlack as text color to work properly on high contrast color schemes.
- Improvement: Introduction of toDisableAutoscrollOnEdit. It prevents a node with a large caption to scroll horizontally when is edited.
- Change: Added test for HandleAllocated to TVirtualTreeHintWindow.AnimationCallback.
- Improvement: Update edit bounds when a node's height is changed and editing is active.
- Change: Partly took back the change for overlay images. VT still must support overlay indices the old way (e.g. for system image lists). Overlay indices >= 15 now use the new mechanism and are drawn without the need to set TCustomImagelist.Overlay.
- Bug fix: Insertion order of nodes was wrong in MoveTo for amAddChildLast.
- Change: removed change lock from worker thread. It isn't used any longer.
- Bug fixes: Mantis bug entries #158, 162-172, 174-191, 192-196, 199, 202, 204, 205, 208, 212, 215, 216, 218, 220, 221, 228.
- Other small improvements.

Version 3.8.3 - 4.0.15 (May - November 2003)

- Bug fix: Initial draw selection with the mouse at the end of large trees (1+ million nodes) started with a huge delay.
- Improvement: Better synchronization of tree windows and the worker thread.
- Change: WM_RELEASEEDITLINK removed. It is sometimes problematic to release the link asynchroniously. Another mechanism is used instead.
- Improvement: check images are now public, to allow to use them for own drawing code.
- Bug fix: using Tree.CheckState[Node] in OnInitNode caused an infinite recursion.
- Improvement: toFullRowDrag introduced
- Improvement: tsCheckPropagation introduced
- Improvement: node selection change with the mouse and modifier keys is now more consistent to Windows standard

controls.

- Improvement: new event OnGetCellIsEmpty
- Improvement: TVTColors.HeaderHotColor(see TVTColors.HeaderHotColor Property, page 519) introduced, default value is clBtnShadow as it was hard wired before.
- Improvement: Auto spring feature. Size changes of the header are evenly spread over all columns, which are enabled for this feature. New options introduced: coAutoSpring, hoAutoSpring.
- Change: Header stream version increased to 3. This was necessary because the new coAutoSpring options increased a column's option size from byte to word (now there are 9 options).
- Improvement: Edit property of TStringEditLink(see TStringEditLink Class, page 316) promoted to public.
- Improvement: ShortenString(see ShortenString Function, page 568) better takes right-to-left contexts into account.
- Improvement: toAlwaysHideSelection introduced. Allows to hide node selections entirely.
- Improvement: toUseBlendedSelection introduced. Allows to have translucent node selections.
- Bug fix: Mantis bug entries #140, 144, 125, 122, 129, 147, 148, 149, 152 - 157.
- Improvement: Mantis feature request #113, toSimpleDrawSelection introduced.
- Improvement: ComputeNodeHeight introduced. Helper method to delegate node height calculation to the tree.
- Improvement: Alt key might be pressed when clicking in the tree. This allows to start drawing the selection rectangle also on node captions and images (which would otherwise start dragging).
- Bug fix: ValidateCache was not always called in ToggleNode when InvalidateCache was used.
- Bug fix: FLastHintRect was sometimes not reset preventing so a new hint to appear.
- Bug fix: Redundant ChangeCheckState in HandleMouseDown removed.
- Bug fix: OnHeaderDblClick was triggered even if the column was set to be unclickable.
- Bug fix: Wheel panning and scrolling was not possible if toAutoScroll was not set. This option has another meaning and should not impact wheel handling.
- Bug fix: VT control could not be set as ActiveControl at design time.
- Bug fix: In method ContentToText it could be that the text contained the separator char as regular character, so it was necessary to wrap the text with quotation marks then.
- Bug fix: Bidi mode and aligment was not correctly considered in UpdateEditBounds when grid extensions were enabled.
- Improvement: Check for nil hint data in TVirtualTreeHintWindow.CalcHintRect(see TVirtualTreeHintWindow.CalcHintRect Method , page 511) just to be on the safe side.
- Improvement: TVirtualTreeColumn.ComputeHeaderLayout(see TVirtualTreeColumn.ComputeHeaderLayout Method , page 492) is now virtual to allow descendants to change the layout.
- Improvement: toFullVertGridLines, vertical grid lines can be drawn over the full client area height.
- Improvement: flickering on column resizing is gone.
- Improvement: System conformal border width calculation for certain tasks.
- Improvement: Animation parameter for TVTHeader.AutoFitColumns(see TVTHeader.AutoFitColumns Method , page 550) to avoid the size animation (default: True).
- Improvement: ParentFont property for the header. Default is False to stay compatible with older tree versions.
- Bug fix: cursor rectangle for spanned columns in normal hint mode was too small.
- Feature: the implementation is now more than 30.000 lines in size.
- Bug fix: Access violation fixed, which was sometimes caused by setting VT to edit mode if the old edit link was not freed yet (because it was still handling a message).
- Improvement: Hint animation now does no longer stop quick switches to new hints.
- Improvement: ParentBackground property published.
- Bug fix: vsAllChildrenHidden and vsExpanded are now removed from a node's state if there are no child nodes anymore

- Improvement: column width limit to 10000 is now only applied on non-NT systems (Win9x/Me).
- Improvement: single letter mode in incremental search is not used if the current node also fits the repeated character.
- Bug fix: correct theme change handling when switching to classic mode.
- Improvement: new event OnMeasureItem, new handling for application driven node heights. TCustomVirtualStringTree.ComputeNodeHeight(see TCustomVirtualStringTree.ComputeNodeHeight Method , page 299) implementation to easy node height computation for multi line nodes.
- Improvement: Header is nil'ed when the tree is destroyed and checked before used in TBaseVirtualTree.Notification(see TBaseVirtualTree.Notification Method , page 225) in order to avoid potential problems accessing an invalid address.
- Bug fix: The cut and copy pending states in the tree and participating nodes were not removed.
- Bug fix: csPaintCopy was not considered when painting (used for TWinControl.PaintTo, e.g. in Form.Print).
- Bug fix: DT_NOPREFIX added for header text output.
- Bug fix: Thread safe check for current tree reference in the worker thread, as it can be reset before it was used.
- Bug fix: Color change for non-standard background colors after all columns were hidden.
- Improvement: new node background erase action (eaNone).

Version 3.6.3 - 3.8.2 (February - April 2003)

- Bug fix: Local tree reference in worker thread is erased when a tree removes itself from the waiter list.
- Improvement: TStringEditLink(see TStringEditLink Class, page 316) public methods are now virtual.
- Change: A couple more methods in the header and columns are virtual now.
- Improvement: Introduction of TVirtualTreeColumnsClass(see TVirtualTreeColumnsClass Type, page 599) and GetColumnsClass in TVTHeader(see TVTHeader Class, page 544). This allows for more customization.
- Improvement: DetermineHiddenChildrenFlagAllNodes, tsUpdateHiddenChildrenNeeded, Optimized flag determination to speed up mass changes of the visiblity state of nodes.
- Improvement: Unicode support for inplace editing by utilizing the TNT controls package. This support is by default disabled and can be made active by enabling the compiler symbol TntSupport.
- Improvement: MoveTo is now allowed with Source and Target being the same node, but only for amInsertBefore/After and child nodes only.
- Bug fix: Mantis bug entry #112, #108, #100, #103, #119
- Improvement: All public images properties changed from TImageList to TCustomImageList. (Mantis entry #110)
- Bug fix: Handling for manipulating columns via index and manual deletion.
- Improvement: Some small additions to aid customizations by descendants.
- Bug fix: GetMaxColumnWidth did not consider if there were vertical tree lines.
- Improvement: The internally used edit control in the tree edit link can be changed now by assigning a new control to the Edit property. The edit link will take over the ownership of the new control then!
- Improvement: Header paint info in advanced custom draw events is now changable (declared as var instead const).
- Bug fix: The number of visible nodes was not updated correctly under certain circumstances.
- Bug fix: Invalid tree data in TVirtualTreeHintWindow.IsHintMsg(see TVirtualTreeHintWindow.IsHintMsg Method , page 512) was used under rare conditions.
- Bug fix: Exception in FindInPositionCache due to invalid position cache data.
- Improvement: VT may optionally use a local node memory manager for node allocations. This will increase allocation speed by about 200% for large trees (so node creation and destruction is about 3 times faster). Small trees do not benefit that much from it, so the node memory manager is disabled by default. See UseLocalMemoryManager for more information.

- Bug fix: State change management used in the worker thread sometimes caused a deadlock.
- Improvement: UpdateScrollBars is now virtual.
- Bug fix: The structure change event was not triggered during ProcessOLEData when nodes were copied.
- Bug fix: Failure to initialize the OLE subsystem does no longer throw an exception. It is a non-critical problem if it fails, only OLE drag'n drop and clipboard operations do not work then.
- Bug fix: Check state changing did not consider the permission of the OnChecking event. Fixing this has the wanted side effect that you cannot change a node's check state if it has a tristate checkbox and none of its child nodes are initialized yet.
- Bug fix: DT_NOPREFIX was not used for single line nodes.
- Improvement: speed up for column erasing
- Improvement: Advanced header custom drawing with the ability to schedule element drawing either by the application or the tree.
- Bug fix: Node rectangle calculation in ClearSelection is wrong.
- Bug fix: all remaining (and fixable) Mantis bug entries fixed.
- Improvement: OnStateChange, DoStateChange, centralized state change method with notification for event sink.
- Improvement: DeleteSelectedNodes is now virtual

Version 3.5.8 - 3.6.2 (December 2002, January 2003)

- Improvement: hint flickering on key press is gone.
- Improvement: Position cache filling is now more fail save.
- Bug fix: Mantis bug entry #75.
- Bug fix: Mantis bug entry #74.
- Bug fix: Mantis bug entry #77.
- Bug fix: Mantis bug entry #82.
- Bug fix: system check images size does not fit.
- Optimization: minimal change in HandleIncrementalSearch.
- Improvement: Full boolean evaluation is permanently switched off as VT heavily relies on that setting.
- Improvement: The buffer for incremental search is now public.
- Bug fix: Column additions now set a column's default properties first before doing default notification handling in order to have them available when updating the header/tree as result of the TCollection.Changed event.
- Improvement: The header font is adjusted according to the system font settings.
- Improvement: Exit code for internal node editor does no longer prevent focus switch to other controls.
- Improvement: Multiline support for node captions. New node state vsMultiline (default: off). Note: This support requires Windows NT (4.0/2000/XP and up) for word breaking. The word breaking feature is not available on Windows 95/98/Me systems.

Version 3.5.1 - 3.5.7 (November 2002)

- Improvement: CanFocus is not virtual in Delphi 4 (-> conditional defination of the override keyword).
- Improvement: Most of the properties for the internal edit control are now public.
- Improvement: Edit control in the standard edit link is now accessible via a protected read only property.

- Improvement: Initialization of global structures is now delayed until the first tree is created. This allows use of VT also in special applications like property sheet extensions.
- Improvement: Updating/Updated pair included in VT.Loaded to avoid design time modification state changes.
- Work around: CM_AUTOADJUST(see CM_AUTOADJUST Constant, page 675) introduced to decouple edit window notification and resizing for Win9x/Me systems.
- Improvement: Reintroduction of automatic exit handling for the internal node editor.
- Improvement: System check and flat check images introduced.
- Improvement: Exchanged 'x' for ' ' as the dummy hint string to avoid showing up a 'x' when using TAppliction.Hint.
- Improvement: The virtual string tree does incremental search independently. Use OnIncrementalSearch if you want to override the default behavior.
- Improvement: VK_BACK can be used in incremental search to return to the previous pattern (deletes the last char in the current pattern and search temporarily backwards).

Version 3.4.10 - 3.5.0 (October 2002)

- License: Virtual Treeview is now released under a double license: MPL or LGPL.
- Bug fix: hit test in other than the main column sometimes returned a check box hit.
- Improvement: new property SelectionBlendFactor. Can be used to adjust the blend effect of the selection rectangle (if it is used).
- Improvement: Painting of node images improved to have it exactly as used in standard controls.
- Bug fix: pressed state for a checked node is now reset if another key than VK_SAPCE is pressed.
- Bug fix: font handling in Print caused wrong output on screen after print.
- Improvement: Ability to link Troy Wolbrink's Unicode aware popup menu added. See VTHeaderPopup.pas for more details.
- Bug fix: vsAllChildrenHidden is now removed from the parent node in AddChild.
- Work around: focus changes between VT and wrapped non-VCL controls like TWebBrowser should be accompanied by resetting the ActiveControl property of the tree's owner form.
- Improvement: Consideration of drag objects not derived from the base control drag object.
- Improvement: Keyboard handling for expand/collapse extended to main keyboard (formerly only numpad).
- Improvement: Consideration of the parent form when checking if focusing of a tree is allowed (the VCL doesn't this).
- Work around: When used in a package the special hint window is not freed correctly by the VCL, which causes an access violation on shut down.
- Bug fix: Clipboard format enumeration should be sorted by priority.
- Improvement: TVTHeader.CanWriteColumns(see TVTHeader.CanWriteColumns Method , page 551) introduced to allow descendants to avoid writing columns to the DFM.
- Renamed Canvas to TargetCanvas in TVTBeforeItemEraseEvent(see TVTBeforeItemEraseEvent Type, page 604) (for consistency).
- Support for application defined drag objects (VCL drag'n drop only).
- Bug fix: NC border painting considers now client edge too (if border width is > 0 and border style = bsSingle).
- ChangeScale implementation / toAutoChangeScale, This is used for big fonts to scale the default node height automatically.
- Text alignment is preserved in DrawTextW(see DrawTextW Function, page 566).
- WM_THEMECHANGED also wrapped with ifdef ThemeSupport.
- More default values added.

- Tree states property is now writable. Writing to it will not trigger any action, but can be used by descendants.

Version 3.4.1 - 3.4.9 (August - September 2002)

- Bug fix: Delphi Gems Issue Tracker #41.
- Bug fix: Delphi Gems Issue Tracker # 38, The MDI problem work around code in TBaseVirtualTree.WMKillFocus was removed as the problem it was to fix does no longer appear but another problem was created by it.
- Bug fix: The tree options were freed in the tree's destructor but used again afterwards (in Clear).
- Bug fix: inherited call in TBaseVirtualTree.Notification(see TBaseVirtualTree.Notification Method , page 225) included.
- Selection with Ctrl-klick is handle the same way as Explorer does it (selection on mouse up instead down).
- Added reset for last searched node (incrementals search) when the search timer is deactivated.
- Work around problems with keypresses while doing hint animation in IsHintMsg
- Change in Animate, use Cardinal instead Integer.
- Bug fix in ScrollIntoView, scrollbar visibility was not correctly tested.
- Bug fix in WMKillFocus, if toGhostedIfUnfocused is used then the focused node should be redrawn too.
- Bug fix in CopyTo, if user canceled node copy then result is nil now.
- Correction, NewParent in TVTNodeMovingEvent(see TVTNodeMovingEvent Type, page 624) and TVTNodeCopyingEvent(see TVTNodeCopyingEvent Type, page 624) is now Target, because the attach operation might have been a sibling action, where NewParent would be inappropriate.
- Added all possible default values to TVirtualTreeColumn(see TVirtualTreeColumn Class, page 485).
- Drop effect support for VCL drag'n drop.

Version 3.3.3 - 3.4.0 (July 2002)

- Delphi 7 compatibility.
- Bug fix for clipboard formats. The internal clipboard formats array was erronously never used.
- Bug fix for freeing image lists if they can get destroyed before the tree.
- Bug fix for ChildrenOnly in IterateSubtree, if the given node has no child nodes.
- Introduced NodeParent property in Virtual Treeview to ease navigation and manipulations.
- Improved client area invalidation check.
- New paint option introduced (toGhostedIfUnfocused).
- New option toDisableAutoscrollOnFocus introduced, to prevent a tree from scrolling horizontally after a column received the focus, but was not fully visible.
- GetTotalCount does not use BeginUpdate/EndUpdate but simple increment/decrement of FUpdateCount to avoid recursion problems.
- DetermineHitPositionLTR and DetermineHitPositionRTL are now virtual.
- PaintCheckImage, PaintImage, PaintNodeButton and PaintTreeLines are now protected (instead private) and also virtual. This will allow for even further customizations of VT.
- Check for FSelectionCount > 0 in RemoveFromSelection to improve stability.
- toReadOnly introduced.
- SetItemHeight renamed to SetDefaultNodeHeight.

- TVTHeader.Invalidate(see TVTHeader.Invalidate Method , page 554) promoted to public.
- Update lock for DeleteChildren operations to avoid access to invalid pointers under certain circumstances.

Version 3.2.0 - 3.3.2 (May - June 2002)

- Fixed hit determination bug (appeared when using margins in the tree).
- Support for Visual Form Inheritance (VFI) for the header.
- Bug fix for loading nodes from stream which are invisible but their parent is expanded.
- Improved theme support. Now TThemeServices from the Windows XP Theme Services (another free software from Delphi Gems) is used. You must now explicitely add a manifest to your application! This is no longer done automatically by the tree.
- Bug fix: autoscroll in VCL drag mode.
- Bug fix: shifted characters for incremental search.
- VST lets now first the ancestor/application render to clipboard before it tries itself.
- Application might modify TargetCanvas.TextFlags in OnPaintText to control the output of normal and static text (currently background only).
- Correct bidi mode window styles.
- Bug fix regarding vsAllChildrenHidden node state (DetermineHiddenChildrenFlag).
- Bug fix in NC painting (removed child window clipping).
- Bug fix horizontal scrolling (ScrollIntoView). Improved horizontal scroll into view.
- InternalConnectNode and InternalDisconnectNode are protected now.
- InitNode in GetHitTestInfoAt to avoid access to uninitialized nodes under certain circumstances.
- Default node text is only stored if it differs from 'Node'.
- Printer font assignment fixed.
- Bug PaintTree for OnPaintBackground fixed. The owner draw mode is now called with the correct window origin set.
- New event OnHeaderDraggedOut.
- Switch to minor version 3.2.
- Hide selection in full row selection mode.
- bug fixes
- other small changes

Version 3.0 - 3.1.9 (January - April 2002)

- First public beta version of the Virtual Treeview CLX version.
- DetermineNextCheckState is now protected and virtual.
- Tree printing.
- UpdateAction only if tree is focused.
- Consideration of the user setting for wheel scroll lines.
- Limit drag over node hits for report mode (like listview).
- All column indexes are now consistently using TColumnIndex(see TColumnIndex Type, page 591) (instead Integer).
- Minor changes to make custom implementations of auto column resize possible.

- Wheel panning and auto scrolling, option toWheelPanning.
- vsClearing node state for optimizations.
- Update*Scrollbar methods are now public.
- toAutoAcceptEditChange.
- MoveTo within a tree now keeps focused node instead resetting it.
- WMContextMenu cancels now also drag operations.
- PaintTree is now public.
- WM_CANCELMODE included.
- Bug fix: IStream storage format does not work with OLEFlushClipboard -> had to remove it (HGlobal is still available).
- Other bug fixes.

Version 2.7. build 2-6 (December 2001)

- child controls are now correctly scrolled too if there is a background image
- tree cursor is now only applied when there is no global cursor (Screen.Cursor) is set
- prevented resize of the edit when grid extensions are active
- selection anchor setting when the first selected node is set in code
- compiler switch ReverseFullExpandHotKey introduced
- Renamed CreateEditor to DoCreateEditor to be consistent with similar methods (DoCreateDataObject
- drastically simplified auto expand code, it also works now as in Explorer
- space handling limited to nodes which have a check box/radio button and if check support is enabled, otherwise space characters are used for incremental search
- change events rework
- ScrollIntoView allows now for vertical centering, option toCenterScrollIntoView
- help contexts for exceptions, EVirtualTreeError(see EVirtualTreeError Class, page 87) now in interface section to allow testing for it in apps.
- ResetRangeAnchor
- VT allows now two storage formats for drag'n drop and clipboard transfers (HGlobal and IStream). Default format is IStream as it does not need as much memory during construction as HGlobal. It is also a faster in usage.
- implementation of events in IDataObject (advise/unadvise sinks etc.) using IDataAdviseHolder
- overloaded GetNodeAt variant which only takes X and Y (in client coordinates)
- ILC_COLOR32 for image lists is only used for Windows NT systems, this will help avoiding GDI trouble on Win9x/Me
- small changes
- bug fixes

Version 2.6, build 3-14, Version 2.7.1 (November 2001)

- F2 alone makes the tree going into edit mode, no longer any modifier key allowed
- added Canvas.Lock/Unlock in PaintTree
- added TDragControlObject assignment in CMDrag
- further small changes for BCB compatiblity

- drag imager helper interface support included thank Jim Kueneman's excellent preparatory work
- structure change event trigger in AddChild
- some minor optimizations
- initial check state setting when changing a check box type
- fmTransparent (button fill mode)
- correct tree window border for themes (still flickers a bit, need any documentation for this)
- theme style is cached now to speed up frequent checks
- improved editing (default editor behavior), correct frame for themed application
- custom checkimages work now also with a themed tree
- OnGetCursor, OnGetHeaderCursor, TVTGetCursorEvent(see TVTGetCursorEvent Type, page 613), TVTGetHeaderCursorEvent(see TVTGetHeaderCursorEvent Type, page 613), DoGetCursor, DoGetHeaderCursor
- changed coMovable to coDraggable, (it was never used so far) and made it actually working
- published Action property
- categorisation of properties for the IDE
- toAutoDeleteMovedNodes
- visible count bug fix
- improved header rect determination and usage
- reset of hot node if focused node is changed
- check button improvement for XP styles
- small tree painting rework
- TColumnIndex(see TColumnIndex Type, page 591), TColumnPosition(see TColumnPosition Type, page 591) (to utilize better type checking)
- overloaded ColumnFromPosition variant to get a column index from a position index
- TVTHeaderDraggedEvent(see TVTHeaderDraggedEvent Type, page 617), new parameter in OnHeaderDraggedEvent
- scrollbar reset when hiding it
- Ctrl-A now considers selection constraints
- no image blending if the tree is unfocused
- improved VCL drag handling
- HasPopupMenu
- other small changes
- bug fixes

Version 2.5, build 39-40; 2.6, build 0-2 (October 2001)

- Release candidate 2 for the beta testers and early adopters
- Full Windows XP theme support
- Legacy code included for XP style support on non-XP systems (TCheckImageKind(see TCheckImageKind Enumeration, page 632), TVTHeaderStyle(see TVTHeaderStyle Enumeration, page 647), TVTButtonFillMode(see TVTButtonFillMode Enumeration, page 642), XPImages(see XPImages Variable, page 667), DrawXPButton, node buttons)
- Node height bug fix for loading trees from stream
- VCL drag handling improved

- Update blocker in AddChild
- Property DragCursor published
- ILC_COLOR32 is now used for image list creation (instead ILC_COLOR16) to allow for XP alpha blending
- ContentToXXX routines consider now hidden columns
- toFullRepaintOnResize
- Header drop mark is not shown if the column being dragged is also the current drop target
- TBaseVirtualTree.GetHeaderClass(see TBaseVirtualTree.GetHeaderClass Method , page 206) (allows creating an own header class)
- Correct space distribution for centered column headers showing also the sort indicator
- Reset of FRangeAnchor when node is deleted
- Conditional compilation of flat scroll bars (see symbol UseFlatScrollbars)
- Synchronous update mode (BeginSynch, EndSynch, tsSynchMode, usBeginSynch, usSynch, usEndSynch)
- toReportMode in TreeOptions.MiscOptions, to even better simulate TListView
- TVTDropMarkMode(see TVTDropMarkMode Enumeration, page 645) for header custom draw
- Other small changes
- Bug fixes

Version 2.5, build 23-38 (September 2001)

- Windows XP style check images
- more available check images
- MDI child parent form problem work around in TBaseVirtualTree.WMKillFocus
- check for destruction of the header popup
- published OnContextPopup
- stop draw selection mode before inherited mouse button up handler opens a popup menu
- corrected some spelling errors
- SetVisible improvements
- FullCollapse changed again, it does not initialize nodes anymore
- CanShowDragImage is now virtual
- changes to provide a drag image of the tree without showing it (for descendants which have own image handling)
- conditional focus setting
- GetFirstVisibleChild(NoInit), GetNextVisibleSibling(NoInit), GetPreviousVisibleSibling(NoInit)
- VisiblePath now checks for vsVisible style and sets it if VisiblePath is set to True
- bug fixes in visibility setting
- toAutoHideButtons auto option
- vsAllChildrenHidden node flag
- VCL drag image bug fix (external drag images)
- small improvement in DrawTextW(see DrawTextW Function, page 566)
- bug fixes background painting
- bug fixes VCL drag image painting (for external drag images)
- changed OnDrawHeader to OnHeaderDraw to fit it closer to the other header events

- shadows for hints and tooltips
- Windows XP style header drawing
- TVTButtonFillMode(see TVTButtonFillMode Enumeration, page 642), ButtonFillMode
- alpha blended selection rectangle
- properties DrawSelectionMode, SelectionRectangleBlendColor and SelectionRectangleBorderColor
- OnHeaderDragged published
- removed TVTEdit.WMKillFocus
- TCustomStringTreeOptions(see TCustomStringTreeOptions Class, page 250)
- adjustments so that TCustom... trees only use and return TCustom... options versions
- other small changes
- bug fixes

Version 2.5, build 1-22 (August 2001)

- removed TVTEdit.WMKillFocus
- TCustomStringTreeOptions(see TCustomStringTreeOptions Class, page 250)
- adjustments so that TCustom... trees only use and return TCustom... options versions
- hint positioning
- tree options are now really overrideable and extendable
- IsVisible[Node] := True now makes a node really visible (expands all parent nodes)
- significant speed improvements for ContentToXXX routines
- better Delphi 6 compliance
- EndUpdate does nothing if the tree is being destroyed
- double click on state icon does toggle node too
- InvalidateNode checks now for allocated handle
- GetMaxRightExtend now correctly includes FMargin in entire width
- DoCanEdit, GetImageIndex (separated from DoGetImageIndex), DoGetText called by GetText
- improved key conversion for incremental search
- support for standard actions
- options splitted into sub-options, property Options is now a class instead of a set
- new options toUseBlendedImages and toAutoScrollOnExpand
- DoBeforeCellPaint is now called in PrepareCell to allow customization after column color application
- consolidated DoDrag* and Drag* methods, DoDrag* methods only call their appropriate events
- AddChild and InsertNode can now take a pointer to user data which is placed into the first four bytes of a node's user data area (there must of course at least be 4 bytes user data).
- vsInitialUserData to indicate a node needs OnFreeNode even if it is not "officially" initialized
- FDragSelection is now also a protected property
- LineMode
- ContentToRTF improvements for correct table building
- ContentToHTML improvments and bug fixes
- changed CF_RTF* to CF_VRTF(see CF_VRTF Variable, page 659)* to avoid identifier conflicts

- internal data handling improved, method AllocateInternalDataArea, method InternalData
- improved text painting
- rounded selection rectangles, property SelectionCurveRadius
- selection border colors
- hatSystemDefault, DoGetAnimationType
- small changes
- bug fixes

Version 2.4, build 1-34 (May to June 2001)

- introduced build numbers
- Delphi 6 compatibility
- brush alignment bug for dotted lines fixed
- test for TYMED_HGLOBAL is now done using a mask instead of direct comparation
- tree column classes can now be changed by descendants, see TBaseVirtualTree.GetColumnClass(🡅 see TBaseVirtualTree.GetColumnClass Method , page 204)
- TVTRenderOLEDataEvent(🡅 see TVTRenderOLEDataEvent Type, page 626), property OnRenderOLEData, DoRenderOLEData TVTGetUserClipboardFormats, DoGetUserClipboardFormats, property OnGetUserClipboardFormats
- removed ScrollIntoView from AddChild and InsertNode
- property OnPaintBackground, DoPaintBackground
- TVirtualTreeColumn.LoadFromStream(🡅 see TVirtualTreeColumn.LoadFromStream Method , page 494) adjustments for the new header stream version
- BeginDrag is again public, TControl already has this method in the public section
- GetFirstSelected and GetNextSelected iterate now through the tree as every other of those methods returning so the nodes in logical order
- GetFirstCutCopy, GetNextCutCopy
- ContentToRTF, ...HTML, ...CSV, ContentToClipboard
- GetFirstInitialized, GetNextInitialized, GetPreviousInitialized, GetLastInitialized
- on expanding scroll child nodes into view
- new property editor for clipboard formats
- procedures RegisterVTClipboardFormat(🡅 see RegisterVTClipboardFormat Function, page 568) etc. added
- property ClipboardFormats added
- IDataObject handling and customization added
- trees render their clipboard formats now on their own behalf, IDataObject does only dispatch calls
- OLEFormats property removed
- clipboard handling reworked
- TVTDataObject(🡅 see TVTDataObject Class, page 521), to have two instances (one for dragging, one for clipboard)
- IDataObject is no longer implemented by the drag manager
- renamed TVTMoveRestriction to TVTDragMoveRestriction(🡅 see TVTDragMoveRestriction Enumeration, page 644)
- correct background erasing for animated toggle
- Incremental search included in WM_KEYDOWN handling instead WM_CHAR with proper ANSI to Unicode char conversion.
- OnUpdating event, DoUpdating method

- improved FullExpand, FullCollapse
- improved AutoFitColumns
- header stream version increase
- Color, coParentColor, poColumnColor for columns (streaming and assignment updated accordingly)
- no scrollbar updates anymore in AdjustTotalHeight to avoid unwanted side effects
- Editors can now prevent node edit stop. CancelEditNode, EndEditNode, DoCancelEdit and DoEndEdit are now functions and return True if editing was stopped.
- small changes in ReinitNode/Children
- workaround for an unwanted drop action when dropping while auto scrolling
- IVTDragManager(see IVTDragManager Interface, page 704)
- tsNeedRootCountUpdate
- WM_NCRBUTTONDOWN in header
- change of focused column with hot keys in grid mode is now limited to not-full-row-select mode
- checks for update count in ToggleNode
- CM_FONTCHANGED
- SetChildCount/property ChildCount accepts now nil to change the top level node count
- improved GetHasChildren
- incremental search improvements
- GetLastVisible, GetLastVisibleNoInit
- Changed semantic for GetLastChild, GetLastVisibleChild, GetLastChildNoInit and GetLastVisibleChildNoInit. They do not iterate the entire child and grand child list but only the child list of the given node.
- Deeper iteration to grand children is done via GetLast, GetLastNoInit, GetLastVisible and GetLastVisibleNoInit
- customizable line styles
- DoGetPopupMenu
- OnDragDrop event has a changed parameter list (no open array as parameter to avoid trouble with BCB)
- public property Image of TVTDragImage(see TVTDragImage Class, page 529) to have access to the internal drag image bitmap
- TVTNodeAlignment(see TVTNodeAlignment Enumeration, page 652), property NodeAlignment
- incremental search
- key handling for non-grid mode improved
- small improvements
- bug fixes

Version 2.4 (April to May 2001)

- key handling for non-grid mode improved
- voDisableDrawSelection (32 bits are now used for options, can't add any more)
- voHideSelection
- GetLast, GetLastNoInit
- incremental search (TVTSearchDirection(see TVTSearchDirection Enumeration, page 654), TVTIncrementalSearch(see TVTIncrementalSearch Enumeration, page 649), SearchTimer(see SearchTimer Constant, page 690), event OnIncrementalSearch, DoIncrementalSearch, TVTIncrementalSearchEvent(see TVTIncrementalSearchEvent Type,

page 621), TVTSearchStart(see TVTSearchStart Enumeration, page 654), IncrementalSearchStart)

- improved header timer handling
- improved key navigation in grid mode
- Virtual Explorer Tree (VET) written by Jim Kueneman is now part of the package
- VK_HOME and VK_END set now first and last column correctly
- removed ivsVisible style because of unpredictable interferences with other code
- columns store their last width and can restore it
- TVirtualTreeColumn.RestoreLastWidth(see TVirtualTreeColumn.RestoreLastWidth Method , page 495), TVTHeader.RestoreColumns(see TVTHeader.RestoreColumns Method , page 555)
- restore last column widths
- workaround for bad implementation of disabled images in TImageList
- brush alignment for drawing of nodes with odd height
- dotted lines implementation improved, tree lines are now dotted drawn too
- Column parameter in TVTDragAllowedEvent(see TVTDragAllowedEvent Type, page 609)
- flat check images, ckFlat
- InvalidateChildren
- arrow key navigation limited to grid extension, otherwise (extended focus) normal behavior
- VK_TAB handling, WantTabs property
- published OnShortenString in the string tree
- introduced a build number in the main version number
- toggle animation only if not the last visible node to be expanded
- CharCode in OnKeyAction is now a variable to allow changing it
- nodes in SelectAll are now initialized
- Position in TVTPopupMenu event
- tsVCLDragging, tsOLEDragPending, tsOLEDragging
- limited auto scroll to draw selection and dragging
- AutoFitColumns
- public property EditLink
- ProcessMessage in IVTEditLink(see IVTEditLink Interface, page 707)
- improved change handling
- InvalidateColumn
- TVirtualTreeColumns.IsValidColumn(see TVirtualTreeColumns.IsValidColumn Method , page 508)
- draw selection is now also possible with full row select
- OnScroll, DoOnScroll, TVTOnScrollEvent
- scrolling if scrollbar is not visible
- UnselectNodes
- deselection with Ctrl+Shift if last focused node is not selected
- node toggle improvements
- background image offsets as properties
- more BCB adjustments
- animated hints improved

- animated toggle improved
- method Animate (general animation support)
- initial range anchor setting if there was not yet a focused node
- animation duration
- PaintImage improvements for transparent images and full row selection
- function Path
- other small changes
- bug fixes

Version 2.3 (March to April 2001)

- tsIterating state (checks in DeleteNode and DeleteChildren)
- paint optimizations
- selected images are dimmed now
- ShortenString(see ShortenString Function, page 568), DoShortenString, OnShortenString event
- OnKeyAction
- scroll bar improvements
- application defined check image list
- internal data handling
- drag image implementation finished (finally, this was really tough stuff because of the alpha blended image and updates in non client area)
- FormatEtcList in the drag manager is now accessible through a property
- clipboard handling
- GetFirstNoInit (renamed GetFirstNode to GetFirst as it is more consistent)
- small changes in TBaseVirtualTree.DoEdit(see TBaseVirtualTree.DoEdit Method , page 182);
- restructuring of node checking
- high color format for internal image lists
- NewParent in OnNodeCopying
- other small changes
- bug fixes

Version 2.2 (March 2001)

- MMX feature check
- property OffsetXY
- drag image
- improved dragging
- general drag management improvements
- TVTDragImage(see TVTDragImage Class, page 529)
- alpha blending

- Watcher(🏠 see Watcher Variable, page 666) (critical section) introduced
- MMX AlphaBlend(🏠 see AlphaBlend Function, page 565) implementation
- improved image painting (ghosted, overlay etc.)
- hoDblClickResize
- TVirtualTreeColumns.AnimatedResize(🏠 see TVirtualTreeColumns.AnimatedResize Method , page 502)
- column resize on double click
- GetMaxColumnWidth
- poDrawFocusRect, poDrawSelection in paint options
- ChildNodesOnly in IterateSubtree
- OnColumnClick, OnColumnDblClick
- HandleMouseDblClick, WM_RBUTTONDBLCLK, WM_MBUTTONDBLCLK
- TVTDragDropManager.SetOLEFormats is now overridable
- hint positioning
- reset of node widths on main column switch
- optimized tree and header painting
- edit mode for item clicks beside the label when grid extensions are set
- TVirtualTreeColumn.GetAbsoluteBounds(🏠 see TVirtualTreeColumn.GetAbsoluteBounds Method , page 493)
- tsPainting state
- simple DrawTextW(🏠 see DrawTextW Function, page 566) implementation (works also on Win9x)
- improved selection rect painting
- tsValidationNeeded
- check event rework
- PrepareGridExtensions
- CM_ENABLEDCHANGED for desgin time
- public header click index
- draw selection improvement for all text alignments and bidi modes
- more header mouse events (OnHeaderDblClick, OnHeaderMouseDown, OnHeaderMouseMove, OnHeaderMouseUp)
- virtual event trigger methods for those mouse events
- multiline hints
- THitPositions(🏠 see THitPositions Type, page 594)
- other small changes
- bug fixes

Version 2.0 to 2.1 (January to February 2001)

- improved hinting (accounts now for alignment and directionality)
- improved GetDisplayRect
- FNodeCache removed
- BidiMode in OnDrawNode
- DetermineHitPositionLTR, DetermineHitPositionRTL

- improved GetHitTestInfoAt
- improved GetNodeAt method
- FindInPositionCache
- made the header the sender in all events related to the header (e.g. OnHeaderClick)
- WM_PRINT, WM_PRINTCLIENT
- Text property for TVirtualStringTree(see TVirtualStringTree Class, page 402)
- corrected header painting for various border style combinations (WS_BORDER, WS_THICKFRAME, WS_EX_CLIENTEDGE)
- check for recursive hint animation loop entrance
- voPopupMode
- Tree.Assign
- Ctrl-A handling (select all)
- context menu key handling (popup menu)
- DoPopupMenu
- right-to-left drawing
- some more adjustments for C++ Builder
- improved column auto sizing and recover for zero-sized columns
- columns can now be used even if the header is invisible
- column autosizing and hints while editing
- GetNodeAt can now take absolute and relative coordinates
- TWMContextMenu(see TWMContextMenu Type, page 630) declaration for Delphi 4
- TChangeReason(see TChangeReason Enumeration, page 631) for OnStructureChange
- hint window improvements for RTL columns and user defined fonts
- drag manager referencing redesigned, no explicit reference count modifications necessary anymore
- complete paint restructuring, now there is only one method to paint the tree: PaintTree, which can be used for normal paint, printing, drag image etc.
- TVirtualTreeColumn.GetDisplayName(see TVirtualTreeColumn.GetDisplayName Method , page 493) changed to show a column's name in the property inspector if it only contains ANSI characters
- column alignment and bidi consideration, added general property in TVirtualTreeColumn(see TVirtualTreeColumn Class, page 485)
- properties IsVisible (changed semantic), VisiblePath, FullyVisible
- filtered IterateSubtree
- WM_CONTEXTMENU
- vsVisible, full visibility implementation for individual nodes (see also GetVisibleParent, GetNextVisibleNoInit etc.)
- included FlatSB.pas in uses list to use the flat scrollbar wrapper in case there is a system not providing flat scrollbars
- ShortenString(see ShortenString Function, page 568) speed improvements
- introduced stream version for header
- OnCreateDragManager event
- EditDelay property
- dropmark can now be switched off (voShowDropmark)
- tree colors class which unites all customizable colors into one tree property
- sort enhancements (auto sort option, sort column, sort direction and sort glyph in header)
- new tree states for left, middle and right mouse button presses

- VCL drag'n drop is now also possible (left mouse button only)
- many minor changes
- bug fixes

Versions 1.30 to 1.31 (December 2000 to January 2001)

- adjustments for C++ Builder (some type declarations moved)
- voSiblingSelectConstraint
- full MainColumn implementation, the column containing the tree can now freely be chosen
- DragOperations property
- AutoExpandDelay
- header image list
- canvas font change tracking during paint cycles
- WM_ENABLE
- Tag property for a column
- OnAfterCellPaint, OnBeforeCellPaint
- item customization reworked, TDrawInfo as well as OnGetDrawInfo is no longer needed
- InitNode in DoGetText
- BeginDrag is now protected and should no longer be used by applications
- many minor changes
- bug fixes

Versions 1.22 to 1.29 (November 2000)

- many other minor adjustments
- OnDragAllowed for selective drag start
- edit improvements
- VK_MULTIPLY handling
- TScreen.HintFont(🏠 see HintFont Variable, page 661) replacement and Rectangle() version for Delphi 4
- column options
- bug fixes

Version 1.21 (Oktober 2000)

- header drag mark improvements
- utility images (internal use, e.g. for header drag mark)
- node focus change events
- column options
- collapse/expand animations

- hint animations
- paint improvements
- splitted stream and tree versions
- property IsDisabled
- bug fixes

Versions 1.17 to 1.20 (September 2000 to Oktober 2000)

- single scroll bar properties class
- property IsDisabled
- separate tree and stream versions
- bug fixes

Versions 1.5 to 1.16 (August 2000 to September 2000)

- small improvments
- InternalAddFromStream, AddFromStream
- grid and tree line colors
- improved constrained selection
- bug fixes

Versions 1.8 to 1.14 (June 2000 to August 2000)

- header streaming
- header hints
- header drag events
- node button
- wide string streaming support
- margins
- gridline color
- improved drag image handling
- generic editing
- non-client area clipping
- worker thread improvements (for use in DLLs, services etc.)
- initial help file and preparation for first public release
- hit test for spanned columns
- clipboard and drag'n drop improvements
- header owner draw
- node sorting (merge sort)
- bug fixes

Versions 1.6 to 1.7 (May 2000)

- initial expand state (ivsExpanded)
- node sort
- MarkCutCopyNodes
- InitChildren, ValidateChildren
- improved clipboard handling (WM_CUT, WM_COPY, WM_PASTE and more)
- voInitOnSave
- overlay images
- no width cache anymore, GetMaxRightExtend instead
- column resize event
- header popup menus, custom draw, dragging, switchable images
- new tree states (expanding, collapsing, updating)
- header options
- auto span coulumns
- bug fixes

Version 1.5 (April 2000)

- own implementation for scroll bars
- background image
- improved NC painting
- improved hit test
- new events and methods (OnNodeCopying, ReinitNode etc.)
- generic node edit improved
- property vsHasChildren for nodes, properties ChildCount, ChildrenInitialized and HasChildren in tree
- header painting improved (is now also double buffered)
- node hint improvements
- improved/extended column handling (hit test, FocusedColumn, voExtendedFocus, GetNext(Visible)Column, GetPrevious(Visible)Column, improved autoexpand, voAutoSpanColumns)
- custom draw (paint cycles introduced: On(Before/After)[Item]Paint)
- many other small improvements and bug fixes

Version 1.4 (February 2000 to March 2000)

- node editing, IVTEditLink(see IVTEditLink Interface, page 707), WM_RELEASEEDITLINK, application defined node editor
- streaming, tree virtualization, application driven save and restore nodes
- OLE clipboard support

- OLE support by the tree to simplify the work the app. must do (ProcessDrop, ProcessOLEData)
- switchable OLE formats the tree should provide
- tooltips, node hints
- GetSortedSelection, GetSortedCutCopySet
- improved accessibility like: TopNode, auto expand/collapes on node focuse change, DeleteSelectedNodes, SelectAll, InvertSelection, GetNext(Previous)Sibling
- splitted change event into two, one for node focus change and one for structure change (Add, Delete etc.)
- disabled nodes, disabled tree, cutted nodes (with proper visual feedback)
- column images
- hideable columns, columns auto sizing and reordering
- normal and static text for captions
- general optimizations
- application driven help and popup menu
- compatibility for Delphi 4
- bug fixes

Versions 1.2 to 1.3 (January 2000)

- various new navigation functions (GetFirstSelected etc.)
- VCL and OLE drag'n drop united (accept only) plus some support routines (e.g. MoveTo, ConvertSubTree)
- new options (auto tristate, auto focus etc.), constrained selection
- header and columns (plus support functions)
- crossed 10.000 code lines boundary on 31. January
- bug fixes

Version 1.1 (December 1999)

- OLE drag'n drop
- check support
- multi selection and other optimizations
- bug fixes

Version 1.0 (July 1999 to November 1999)

- base implementation (buttons, lines, general window handling, base mouse handling)
- caching for optimal speed
- multi selection
- OLE drag'n drop
- common tree functions and properties (InvalidateNode, GetNodeData, Visible, Selected, Expanded...)

5 Inner fundamentals

If you would like to understand certain things going on in Virtual Treeview you should read on in this chapter.

Description

Herein I will explain what was my motiviation to create Virtual Treeview, why I based everything on the design you see now, how painting in the tree is organized and can be customized, what are the key navigation commands in the control and many more things. Particularly if you want to derive your own descendant from Virtual Treeview you should read everything here.

Topics

5.1 The virtual paradigm

Interested in the story of Virtual Treeview? Well, here is a part of it.

Description

The History

Years ago I wrote a treeview implementation called TreeNT (see also TreeNT at the Delphi Gems homepage). This control is a wrapper around the system tree control provided by ComCtl32.dll. Over the time while I developed the control I encountered many limitations, either introduced by the Delphi VCL or "intended" by the underlying system control. The most annoying problems were the dependency on specific ComCtl32.dll versions and the slow behavior of the control when more than a couple of nodes had to be managed. In fact Microsoft's tree view has been designed to ease life for small node sets only.

The problems

Despite the problems with the system tree control TreeNT worked quite well and has meanwhile been downloaded several thousands of times from my web site and those many other Delphi sites around the world. When I started working for a software house in Munich I quickly included TreeNT into the company's inhouse library. But then the problems which were formerly only annoying started to make the tree nearly unusable. I realized how much the requirements in the private and professional/commercial environment actually differ.

Aside many other problems one was especially annoying: How can adding some 5000-6000 nodes take a minute or so to finish? This question was the reason that I created the very first version of Virtual Treeview. What I actually did was to recall my studies where I learned my trade. Why, on earth, must everything be wrapped into an object? In Java and the like even simple data types like strings are objects. While this kind of abstraction provides some additional conveniences it costs quite a lot in terms of CPU power and memory, particularly if it comes to many instances of such simple type pretenders.

The nodes

These thoughts inspired the idea of using small records as nodes only and putting them into a doubly linked list (see also TVirtualNode(see TVirtualNode Record, page 576)). Well, this idea is not very new (in fact I used to write many code parts using linked lists), but together with other principles it got a new quality. The key points are

- node minimalism and
- pull over push.

Pull over push means here that the tree asks for the data it must display instead of having the application to push it into the tree during creation. A node stays uninitialized and dataless until it is touched the first time. Only its existence and place in the tree is known. The assumption that this would be much better in terms of speed and responsiveness was based on the thought that only very few nodes need really to be accessed usually (mainly to display a handful of nodes in the tree window). Tests confirmed quickly that this was indeed the case.

The node minimalism lead to the approach to leave out everything from the node structure which can be determined dynamically and/or is used very rarely. One example is the owner tree of the node. There are only very few cases where the knowledge about it is necessary. So a standalone method (TreeFromNode(see TreeFromNode Function, page 569)) has been created to allow retrieval of the owner tree. Another omitted member was the absolute position of a node which is needed e.g. for invalidation of a certain node or start of tree window painting. For this decision however another fact was more relevant: inserting, deleting, collapsing, expanding and hiding nodes makes all following positions obsolete and requires a rescan and update of the tree. Since this would be much too expensive a node cache has been introduced. This cache is a simple one-dimensional array which holds node references in increasing absolute position order. A separate thread (which is shared between all Virtual Treeview instances in a program) is used to collect the references in the background. Well, one could say that all these updates are still necessary (even with a cache because it must be held coherent) and the thread could well work directly in the node records. The most valuable advantage of the array like cache is however that you can query it for a node at a particular position by using binary search which is not possible with linked lists.

The paradigm

Being virtual is more than requesting data on demand. Although this is an important aspect some additional things are

considered in Virtual Tree. The pull over push principle for data can be extended for the structure as well. It means then to create nodes or entire branches only on demand (e.g. when expanding a node or iterating through its child nodes for incremental search etc.). This allows to fill a tree view with only the top nodes and initialize only those of them which are currently in view. Clearly this increases start up times a lot for large trees.

The core sequence for filling the tree is an iteration, which runs over initializing a node (to tell if it has children at all, see OnInitNode) and initializing its children (see OnInitChildren), which only means to tell the tree how many child nodes should be there. The tree will automatically allocate memory and set up the structure in the most efficient way but does not yet query for data. This will then again be done in OnInitNode for each of the newly created child nodes as soon as they are touched the first time. For compatibility reasons also AddChild and InsertNode have been implemented but are not as efficient as the iterative approach just explained. For obvious reasons these compatibility methods have to trigger some updates for the tree implicitly unless updates are locked. It is therefore strongly recommended to put calls to AddChild and InsertNode always into a BeginUpdate/EndUpdate frame (if there is more than one call).

Records instead classes

Basically, the idea of virtualizing the tree control and using records instead of classes were two ideas which are born nearly at the same time. It was quite clear from the very first moment that classes can never be as effective as a simple record structures (in terms of size, access speed and management). Sure, a TPersistent only needs 4 bytes more than a record (the pointer to the class' VMT), but these are still too many extra bytes if you consider that I have wrestled quite a while with myself about every byte in a tree node (and want the minimalism principle). Another point you should not underestimate is that classes as nodes would of course also mean to put node specific methods into this class too, which will be overridden at times (this is the main argument to use a class after all). This will require additional CPU cycles just to lookup access methods, to dereference etc. which in turn will cost extra time. Trees with only some 1000 nodes will never see a large difference but for big trees this is significant and Virtual Treeview has mainly been created to address high capacity tree views.

With choosing records I also gave up the VCL concept of having a tree nodes class which is responsible to manage tree nodes and is secondary to the control itself. In Virtual Treeview every access to the tree content is done via methods and properties provided by the tree control. Keep also in mind that nobody prevents you from using classes and store their references in the node's data area. It is only just so that the node (as internal management structure) is as small as possible, opening so all possibilities: from smallest memory footprint to highest comfort.

19.09.2003

Times are changing

With the advent of .NET and C# things outlined in the previous paragraphs need rethinking. The software world is changing and so must Virtual Treeview if it wants to stay. Don't get me wrong, all the nice principles in the control have proved their usefulness and fitness for the purpose they were designed. However one could see that there are still flaws and probably will ever be, regardless of the actual design. Still, nothing is so good that it couldn't get better and the approach using records/structs instead of classes not only made it sometimes hard to get used to Virtual Treeview but it makes the control as a whole incompatible to the intrinsic values of Microsoft's new concept. And here lies the next natural step for it: Virtual Treeview must go .NET. So stay tuned for the things to come...

Group

Inner fundamentals(see page 33)

5.2 Paint cycles and stages

The most complex process in Virtual Treeview is without doubts its painting. Read here what stages Virtual Treeview enters during paint and how you can customize this process.

Description

Similar to the system tree view Virtual Treeview defines so called paint cycles. A paint cycle is one run of the paint code which draws a part or the entire window. In Virtual Treeview this task is accomplished by the method PaintTree which centralizes the paint management into one place and is called for various tasks like window painting, drag image painting, WM_PRINTCLIENT handling and so on.

This paint method is able to draw the entire tree regardless of its window to the target canvas and optimizes painting by considering the update/clipping rectangle, which is passed in via the Window parameter (see also PaintTree).

Usually the following paint stages are executed during a paint cycle:

- before paint (OnBeforePaint)
- before item paint (OnBeforeItemPaint)
- before item erase (OnBeforeItemErase)
- after item erase (OnAfterItemErase)
- before cell draw (OnBeforeCellPaint)
- on paint text (string trees only, OnPaintText)
- after cell draw (OnAfterCellPaint)

after item paint (OnAfterItemPaint) after paint (OnAfterPaint)

The cell and node events are of course not executed if there is no node to be drawn. A special flag (tsPainting) in TreeStates indicates when a paint cycle is in progress. Using this flag an application can for instance determine whether a node is initialized because it is about to be drawn or for other reasons.

Every of the stages above is accompanied by a specific event which allows the application to customize a particular aspect in the painting. The following list discusses tasks which can be done during the various stages.

Stage	Description	Comments
before paint	This stage is entered only once per paint cycle. After setting the vsPainting state it is the very first instruction in a cycle.	This stage is typically used to do any further setup of the target canvas of the paint operation (e.g. the window or a printer canvas), like changing the mapping mode or setting another clipping region. Since the passed canvas is not directly used to do the actual painting setting its font or colors has no effect. Basically only properties which affect blitting a bitmap to the target canvas have an effect at all.

before item paint	This stage is entered once per node to be drawn and allows directly to control the path which is the taken to paint the node.	In the event for this stage you can tell the tree whether you want to paint the node entirely on your own or let the tree paint it. As this happens on a per node basis it is the perfect place to maintain a special layout without doing everything in the paint cycle. Note: setting the CustomDraw parameter in the event to True will skip the node entirely, without painting anything of the standard things like tree lines, button, images or erasing the background. Hence to display any useful information for the node do it in the OnBeforeItemPaint event. This is the first stage which gets the double buffer canvas which is used to draw a node so if you want to set special properties this is a good opportunity. Keep in mind though that in particular the colors are set by the tree according to specific rules (focus, selection etc.).
before item erase	This stage is also entered only once per node and allows to customize the node's background.	This stage and its associated event is usually used to give the node a different background color or erase the background with a special pattern which is different to what the tree would draw.
after item erase	This stage is also entered only once per node.	This stage and its associated event is used to do additional drawings after the background has been erased.
before cell paint	This paint stage is the first of the cell specific stages used to customize a single cell of a node and is called several times per node, depending on the number of columns. If no columns are used then it is called once.	While internally a full setup for this node happened before the stage is entered (if it is the first run) the only noticeable effect for the application which has changed compared to after item erase is that the painting is limited to the current column. There are still no lines or images painted yet.
on paint text	After default stuff like lines and images has been painted the paint node/paint text stage is entered.	Because Virtual Treeview does not know how to draw the content of a node it delegates this drawing to a virtual method called DoPaintNode. Descendants override this method and do whatever is appropriate. For instance TVirtualDrawTree(see TVirtualDrawTree Class, page 323) simply triggers its OnDrawNode event while the TVirtualStringTree(see TVirtualStringTree Class, page 402) prepares the target canvas and allows the application to override some or all canvas settings (font etc.) by triggering OnPaintText. After this event returned the text/caption of the node is drawn. Changed font properties are taken into account when aligning and painting the text. Note: The string tree triggers the OnGetText event two times if toShowStaticText is enabled in the TVirtualStringTree.TreeOptions.StringOptions property. Once for the normal text and once for the static text. Use the event's parameter to find out what is required.
after cell paint	This stage is entered immediately after the cell is drawn.	This stage can be used to add whatever you like to a single cell after everything has been painted there and is triggered once per column.
after item paint	This stage is entered after all cells of an item are drawn.	The after item paint stage is used to add node specific stuff like frames and the like which concern all columns of that node and is called once per node.
after paint	The after paint stage is the last stage in the long chain of paint stages and is entered after when paint cycle is complete.	In this stage everything of the tree (related to the current update area) has been drawn, including the selection rectangle.

Group

Inner fundamentals(see page 33)

5.3 Tree image and tree window

If you are one of those developers who want to create tree descendants, which perhaps involve visual changes in Virtual Treeview then you need to know how the control paints itself (as outlined in Paint cycles and stages(see page 36)). What happens with the resulting image and how it can be used for certain tasks like printing? Some answers are in this topic.

Description

Some methods in Virtual Treeview work with an internal tree image, e.g. painting or hit determination. This tree image does not really exist but is rather an imagination of the entire tree drawn to an infinitely sized sheet. In this picture the tree is always drawn at position (0, 0) and advances to positive horizontal and vertical values which reach out to the right and down, respectively. This also means that coordinates given in this fictional image are always positive.

A display function like the WM_PAINT handler can now take a rectangle of this full image (in PaintTree this is called the window) and let it draw to any location in a target canvas. This allows to draw a part of the entire image even if the tree window is scrolled or needs otherwise to be moved (e.g. when dragging or printing). In order to get the full dimension of the tree image call GetTreeRect, which returns a rectangle always starting at (0, 0) and extending at least to client area size but usually much further (determined by the private variables FRangeX and FRangeY which also determine the scroll bar values).

In order to maintain the visual portion of the tree image two offset values are maintained which specify the horizontal and vertical distance relative to the client area of the tree control. These offsets (OffsetX, OffsetY and OffsetXY) are therefore negative. This means 0 means no offset at all and -100 means the tree is scrolled by 100 pixels. Values > 0 are always made to 0.

How does this now fit together when you want, say, to print a part of the tree to a memory or printer canvas? Have a look at the image below:

On the left pane you can see a typical tree view of which only a specific part is visible. This situation is visualized by the non-shaded rectangular region. The right pane shows the reproduction of the visible part to different locations. The entire tree image size corresponds to the internal FRangeX and FRangeY variables of the tree view. When drawing a part of the window the method PaintTree needs to know the size and position of the part to draw. This is given by a TRect structure passed in the Window parameter. For normal screen display this rectangle structure consists of the current scroll offsets (properties OffsetX and OffsetY or OffsetXY for both together given as TPoint) and the size of the client area of the tree control. This rectangle is usually also intersected with the current clipping region to avoid painting parts of the tree which are not invalid.

The place where the image is to be painted is given in the parameter Target. This point specifies the physical location in the target canvas where to draw the content of the region specified by Window. Note that these coordinates are usually (but wrongly) considered as being physical pixels. This might be true for screen or bitmap output but is not for the printer where a single pixel would be much too small. Hence another term is used here: logical coordinates. The actual size of one unit of these coordinates can either be a single pixel but also a millimeter, inch or even some other odd size. The interpretation is determined by the mapping mode of the target canvas (device context, DC) and its window and viewport extents. For more information about mapping modes see the online help or MSDN under *SetMapMode* and for DC extents under *SetWindowExtEx* as well as *SetViewportExtEx*. With the help of mapping modes and window/viewport extents you can greatly customize the outcome of PaintTree. These APIs are usually also used to provide a print preview.

Group

Inner fundamentals(see page 33)

5.4 Data handling

An important aspect of the tree is the handling of data for each node. Read here how Virtual Treeview manages your data.

Description

Usually single items (as in TTreeview and TListView) only have a simple data member which can take a pointer to the

actual data an application maintains for this item in an external structure. This principle can be used with Virtual Treeview too. But the control goes a step further by letting the application decide how much data per node is needed and providing this space implicitly. This way the application is freed from maintaining an extra structure sometimes.

Application view

The core point behind this technique is that the tree has to allocate and deallocate memory for each node anyway. The amount to allocate does not matter with respect to node handling. So it is easy for the tree to allocate some more bytes for the application. To know how much memory is to be allocated there are several ways to tell. Firstly there is the property NodeDataSize which can be set already at design time and describes the required user memory per node in bytes. If you don't know this size (because it depends on a structure which you want to be examined by SizeOf) then simply assign your size in the form creation process to the tree via the NodeDataSize property.

Secondly, use the event OnGetNodeDataSize. This event may occasionally be useful for values which are neither known at design time nor can they be determined at compile time (as the size of a record). The event is triggered when the NodeDataSize property is -1 (which is by default the case). This value will be replaced by the actual data size returned in the event.

Note: If you want to store application data in a node (e.g. the caption) then you must allocate node data as outlined above. If you get an access violation in OLE32.dll then you have likely forgotten to allocate this node data and tried to assign a string.

The allocated bytes per node are an inherent part of the node record and follow the last internal member in the TVirtualNode(see TVirtualNode Record, page 576) structure (symbolized by the Data member). In order for the application to access this memory it needs to map its node data structure to this tree internal memory. To simplify this task the application can use GetNodeData. This method returns the address of the data area in a node record. This address can then be assigned to a local pointer variable (or can be type casted) as shown in the chapter code repository. I strongly recommend that you always use the GetNodeData method to get the data address instead of simply using @Node.Data because a tree class may add internal data to this area which starts then at this address while the actual application data begins a few bytes later.

Tree Control view

Depending on its tasks a tree may need to store data on a per node basis (e.g. TCustomVirtualStringTree(see TCustomVirtualStringTree Class, page 274) keeps the width of a node to allow quick response on DoGetNodeWidth which is used for various tasks including draw selection). Particularly multi selection with the mouse (draw selection) depends on very quick width determination to allow interactivity even with 100,000 selected nodes.

In order to avoid access conflicts between the tree and the application a simple mechanism has been implemented to allow flexible internal node data handling (in addition to the normal node record and application data handling). Following functions have been added to the base tree:

- InternalData
- AllocateInternalDataArea

Note: A tree descendant which requires additional internal data must call AllocateInternalDataArea to register its need.

InternalData is a virtual function which does nothing in the base tree class (returns nil). I recommend to override this method in descendants however and return the address of the internal data for that tree. This address can easily be determined by adding the offset returned from AllocateInternalDataArea to the start of the node record. To make this work you have of course to keep the offset somewhere, just like TVirtualStringTree(see TVirtualStringTree Class, page 402) does.

AllocateInternalDataArea is the function which sums up all requests for internal data and keeps this sum which must be added to each node data offset to return the correct address for user data. Note: call this method only once (e.g. during tree creation) to register the data area you need.

Group

Inner fundamentals(see page 33)

5.5 Editors and editing

Because of the virtual nature of Virtual Treeview editing becomes a difficult issue. Read here what needs to be considered and where you can hook in to allow any editor for a node.

Description

Generally it cannot be said what data a user will edit when he or she edits a node. In the case of the string tree it becomes a lot easier to decide because we have, as the name implies, strings and captions to edit. But this is only a special case and the underlaying edit principle must be flexible to allow editing various different data of a node, including several items instead of only single ones.

Since you cannot generally tell what will be edited the used solution does not assume anything. Instead it delegates the entire process to the application or derived trees via the IVTEditLink(see IVTEditLink Interface, page 707) class. This interface defines some necessary methods which allow interaction between the tree and the editor but the actual editor implementation is up to the edit link (which can of course delegate this task to even another instance like the application). The edit link is responsible for everything including to hide and show the editor, reading the old values of a node and setting the new values etc. The tree only signals some general states like the edit start, end or cancellation.

Editing starts with the protected DoEdit method which may be triggered by the edit timer (which in turn is triggered by clicking again on the focused node), by pressing F2 or by calling EditNode. DoEdit creates an editor (actually only the edit link) via the virtual CreateEditor method which should be overridden by descendant trees to return a valid edit link (as TVirtualStringTree(see TVirtualStringTree Class, page 402) does). Otherwise the method will query the application for an editor link. Actual editing starts only if CreateEditor returns a valid edit link.

After the tree received a valid edit link it initiates communication by calling PrepareEdit which can be used by the link to retrieve the values to be edited using the given node and column. If the edit link returns True in this call another call is initiated by the tree telling the link where to place the editor using the SetBounds method. Finally the tree calls BeginEdit to actually start the edit operation. From now on the edit link is responsible for any further action including passing on key presses like VK_UP and VK_DOWN to select a new node to edit etc. The link must also be aware that editing might be stopped at any time by EndEdit or CancelEdit. Otherwise however the edit link (and its editor(s)) is completely autonomous and can use whatever it considers as being appropriate for the editing task. It isn't even limited to use an in-place editor.

With the class TStringEditLink(see TStringEditLink Class, page 316) you will find a sample implementation used in the string tree to edit single node captions. By examining the used editor (a normal TEdit control) you will find some things which should be handled the same or in a similar way to make editing smooth.

Starting with version 3.8 Virtual Treeview allows to use the TNT controls suite from Troy Wolbrink, which allow to edit node captions with Unicode content. Download the latest package and add its path after installation to your project. Enable the TntSupport compiler switch by changing it from {.$define TntSupport} to {$define TntSupport} and recompile.

Group

Inner fundamentals(see page 33)

5.6 Keyboard, hotkeys and incremental search

Virtual Treeview handles most of the important keyboard actions on its own. Also here you can inject your own handling to modify the control's behavior. Read also about incremental search and how it is implemented.

Description

Particularly key navigation is implicitly handled in various ways. A full list of hot keys currently supported by the tree view is shown below. Note that the control key has precedence over the shift key if both are pressed at the same time. This means that in this case the shift key has no meaning.

The tree view supports the same hot keys as the Windows system tree control and allows to customize key messages to change the meaning of the particular key (see also OnKeyAction). Generally speaking all navigation keys change the current selection if no modifier key (like control or shift) is pressed together with the navigator key. Like the system tree control Virtual Treeview allows to modify the current selection by holding down the shift key and pressing home, page up or any other of those keys at the same time. The control key neither changes the selection nor the focused node but can be used to scroll the tree window.

For special handling a grid mode is supported (see toGridExtensions in Options) which changes (among other things) some key semantics. These changes are explicitly marked in the table below.

Key	Modifier	Function
Home	none	Selects the first visible node (the focused column does not change). This node also receives the input focus. Modifications in grid mode: The focused node does not change but the first visible column is focused instead.
	shift	Moves the focus to the first visible node (the focused column does not change) and includes every visible node, from the previously focused to the newly focused one, into the current selection. Modifications in grid mode: Not the focused node is changed but the first visible column is focused instead. The selection does not change (note: you cannot select several columns of the same node).
	control	Scrolls the tree to the top left corner without change of any selection or focused state.

End	none	Selects the last visible node (the focused column does not change). This node also receives the input focus. Modifications in grid mode: The focused node does not change but the last visible column is focused.
	shift	Moves the focus to the last visible node (the focused column does not change) and includes every visible node, from the previously focused to the newly focused one, into the current selection. Modifications in grid mode: Not the focused node is changed but the last visible column is focused instead. The selection does not change.
	control	Scrolls the tree to the bottom right corner without change of selection or focused node.
Prior (page up)	none	Scrolls the tree window and single selects a node one page up. This node receive also the current focus.
	shift	Like without modifier key but includes a page of nodes into the current selection.
	control	Scrolls the tree window one page up without change of selection or focused node.
Next (page down)	none	Same as prior but one page down instead.
	shift	Same as prior but one page down instead.
	control	Same as prior but one page down instead.
Up	none	Advances the focus from the currently focused node to the previous visible node.
	shift	Advances the focus and adds the newly focused node to the current selection.
	control	Scrolls the tree window one line up. One line is defined as the DefaultNodeHeight.
Down	none	Same as up but one line down instead.
	shift	Same as up but one line down instead.
	control	Same as up but one line down instead.
Left	none	Moves the focus to the parent of the currently focused node and selects it if the current node does not have children or is already collapsed. Otherwise the focus is not changed but the node will be collapsed. In both cases the focused node will be the only selected node afterwards. Modifications in grid mode: If extended focus is enabled (see toExtendedFocus in Options) then the behavior changes to a simple navigation to the previous visible column.
	shift	In opposition to the none-modifier case the expand state of the node does not matter nor is it changed. The focus is advanced in any case and sibling nodes as well as the parent node are added to the current selection.
	control	The tree window is scrolled to the left by the amount pixel given in the indent property.
Right	none	Moves the focus to the first child node of the currently focused node and selects it if the current node has children and is already expanded. Otherwise the focus is not changed but the node will be expanded. In both cases the focused node will be the only selected node afterwards. Modifications in grid mode: If extended focus is enabled (see toExtendedFocus in Options) then the behavior changes to a simple navigation to the next visible column.
	shift	Same as the none-modifier case but the selection is extended with the first child node.
	control	Same as left but the tree window is scrolled to the right.
Back	none	Moves the focus to the parent of the currently focused node and selects it as only node.
	shift	Modifier keys have no meaning for this case.
	control	Modifier keys have no meaning for this case.

Tabulator	none	The tabulator key is a bit special because it is only used with grid extensions to advance from cell to cell. Without modifier the focus changes from left to right and from top to bottom. It is necessary that you enable TAB support by setting property WantTabs to True.
	shift	Same as without modifier key but the focus advances backwards, from right to left and bottom to top.
	control	This modifier has no effect.
F1	none	This function key triggers node specific help support. Via the OnGetHelp event the application is queried for a help context to show.
	shift	This modifier has no effect.
	control	This modifier has no effect.
F2	none	This function key turns the tree view into edit mode if there is a focused node, the tree is editable and the application allows to edit the node.
	shift	This modifier has no effect.
	control	This modifier has no effect.
+ (add)	none	Expands the currently focused node.
	shift	This modifier alone has no effect, but see the following comment.
	control	Pressing the control key together with + will start auto sizing all columns in the tree. If the shift key is also pressed then the whole tree is expanded instead.
- (subtract)	none	Collapses the currently focused node.
	shift	This modifier alone has no effect, but see the following comment.
	control	Pressing the control key together with - will restore all columns to their previous widths. If the shift key is also pressed then the whole tree is collapsed instead.
* (multiply)	none	Expands the currently focused node and all its children and grand children.
	shift	This modifier has no effect.
	control	This modifier has no effect.
/ (divide)	none	Collapses the currently focused node and all its children and grand children.
	shift	This modifier has no effect.
	control	This modifier has no effect.
Escape	none	Stops actions which require a specific state in the tree like editing, mouse selecting, drag'n drop etc.
	shift	This modifier has no effect.
	control	This modifier has no effect.
Space	none	Used only if check support is enabled (see toCheckSupport in Options) and the currently focused node has got a check type other than ctNone. In this case the space key switches the check state.
	shift	This modifier has no effect.
	control	This modifier has no effect.
Apps (menu key)	none	Similar to F1 triggers the apps key popup menus on a node by node basis. For more information see also the event OnGetPopupMenu.
	shift	This modifier has no effect.
	control	This modifier has no effect.

A	none	This is the only "normal" character used as hotkey so far. It has only an effect together with the control key.
	shift	This modifier has no effect.
	control	Pressing 'A' together with the control key will select all currently visible nodes in the tree view.

Incremental search

Incremental search is a commonly used term to describe the effect that the user types some letters while the tree view has the focus and the control will try to locate a node whose caption matches the letters. Because Virtual Treeview does not know what caption a node has it cannot compare the incoming letters and uses therefore again an event to ask the application to do the comparison. By using the lesser of both string lengths and a partial comparison in this event the tree will be able to select also partial matches. Note: Virtual Treeview tries to mimic the UI of the system list view and system tree view as close as possible and uses therefore two modes when searching. One is used when there is no key or only one key pressed and the new key is the same as the already recorded one. In this case the search always starts with the next node and only nodes which match the single new key will be found. This allows to quickly cycle through a number of nodes all matching/beginning with the same letter. The other mode is normal linear search where all key presses are recorded and compared with the nodes in the tree. Whenever the application considers a node as match (it even hasn't to have a caption the same as the search string) this node is returned as new target and focused.

Group

Inner fundamentals(see page 33)

5.7 Drag'n drop and clipboard handling

Virtual Treeview behaves also well when it comes to data exchange with other applications or structural manipulations using the mouse. In both cases the prerred method is using OLE. Read here why and what's behind it.

Description

One important aspect for system integration under Windows is the ability to use OLE (object linking and embedding) to transfer data from and to other applications. Unfortunately this is a dark chapter in Delphi's feature list because there has never been support for either OLE drag'n drop or OLE clipboard handling (until Delphi 6 at least). Instead a proprietary mechanism had been invented which is not at all compatible with the rest of the system.

Drag'n drop

Virtual Treeview supports both kinds of drag'n drop (VCL and OLE) and tries to present a single interface to the application. This means that those (already existing) events which can be reused are used in the process (like OnStartDrag and OnEndDrag). Other events however differ significantly from the VCL variants because of the additional information available during OLE drag'n drop. These events are OnDragOver and OnDragDrop. Read there for a detail description of the parameters. Since in a VCL drag'n drop operation the source is always known as being a VCL control it is relatively easy to determine the participants. This however is not very data-oriented and OLE drag'n drop focuses exactly on this issue. In such an operation a so called data object is passed to the receiver which is a COM interface (IDataObject) and can be used to retrieve the dragged data in various formats.

To accept OLE drag'n drop an application has basically the same steps to perform as always used for VCL drag'n drop plus some extra work to handle the different data coming in during the drop event. Usually there is an event handler for

OnDragOver which tells not only whether dropping is allowed on a particular position but also which effect should then take place. Allowed effects are copy, move and link. This is the first new aspect which is not possible with VCL drag'n drop. As always the real work must be done in the drop event and Virtual Treeview supports processing its own native data format (which is a stream of chunks to represent the tree structure) by a special method called ProcessDrop. Note that this method can only be used for the internal format and does not process other formats like text or images. From this information you can easily conclude that a lot of other formats can be passed around with the mighty OLE drag'n drop mechanism. It is however out of the scope of this help to describe how this mechanism works or to give an overview of possible data formats. Please read the Win32 SDK documentation as it comes with your Delphi copy or browse the MSDN online documents at MSDN online for a detailed description. The only interesting aspect you should keep in mind at the moment is that the data object used in a drag'n drop operation is the same as used for OLE clipboard data. Hence you can share code for handling of both and you don't have to learn different ways or data structures.

Step by step

The typical approach to determine how to handle data during the drop event in Virtual Treeview is as follows:

- If the given data object is nil then the source of the drag operation is the VCL and you have to figure out yourself what and how to process the drop. The other parameters contain also mostly useful data (Effects is set to default values however). Read more details at OnDragDrop.
- With a valid data object you know OLE data is being passed. Check the source parameter to learn whether a Virtual Treeview is the source or something else. Although further processing can successfully be done without this information it is still useful if you want to optimize data transition and source as well as target tree are in the same process (in which case source memory can be accessed from the target tree).
- Loop through the given formats list to find a format you can handle. Since it is recommended to sort this list so that preferred formats come first you can simply accept the first format you find in the array which you are able to handle. With a Virtual Treeview as source usually already the second entry represents the native format (the first is a special reference format which is not useful for an application) and can be passed to ProcessDrop. The native format is registered as CF_VIRTUALTREE(see CF_VIRTUALTREE Variable, page 658) while other typical formats include CF_TEXT or CF_HDROP. Note that, because Virtual Treeview is already OLE drag'n drop aware, you do not need to register its window for accepting file drops. If the user drops files onto a Virtual Treeview window you will get the CF_HDROP format in the format list passed to OnDragDrop.
- Depending on the data formats you might want to take various actions. For the native tree format you will likely want ProcessDrop to handle the data. If you made sure source and target tree are in the same application (process) you can even omit the entire handling and simply call MoveTo or CopyTo.
- If you do not call any tree method or handle the dropped data somehow yourself nothing will happen. No data will be added.

Group

Inner fundamentals(see page 33)

5.8 Additional information

This chapter collects everything else which is important or very helpful to know but which does not justify an own chapter.

Description

- Special care has been taken to wrap every event call by a DoXXX method (e.g. for OnBeforeItem paint there is a protected DoBeforeItemPaint method) which is always virtual to allow descendants to override it and intercept so calls

to events regardless whether there is actually an event handler assigned or not.

- During a locked update stage (entered by BeginUpdate) there will be no updates of the tree nor the selection. If you change the selection in such a stage then it is temporarily accumulated and applied if, during an EndUpdate call, the inner update counter reaches zero.

Borland C++ Builder:

- Define the constant NO_WIN32_LEAN_AND_MEAN in your environment/project options to avoid problems with undefined interfaces.
- The automatic conversion process from Delphi source code to C++ code has unfortunately some bugs. Most of them could be solved by rearranging the Delphi code, but one problem still remains and must be solved manually. The translator does not automatically consider default parameters in functions. The parameters are correctly converted but without the default value. Usually the problem will appear when you try to compile and there is a call of the function with fewer than expected parameters.

Group

Inner fundamentals(see page 33)

6 Virtual Treeview step by step

Often a simple step by step tutorial gets you much faster started than a long list of features and possibilities. This topic describes the basic usage on the basis of a simple project.

Description

Written by Sven H. (h.sven@gmx.at), Revision and translation by Mike Lischke (public@delphi-gems.com)

At the time when this description was created I had not much Delphi knowledge and had not yet read through any of my two Delphi books. But I was quite impatient and wanted to try out what is possible. Although I have some knowledge about object oriented programming and C++ (I have learned something about it during my studies), this project was my first attempt to program in Delphi. It could be that I have not provided the most elegant solutions und I am always open for improvement suggestions. But all principles I demonstrated here do work (at least for me J). I have implemented them in my first project this way. This guidance is made in the first place for programmers who are not yet familiar with Virtual Treeview and will so perhaps have an easier start. If you have questions or suggestions regarding this guidance please forward them to h.sven@gmx.at. For other questions you can contact Mike and use the dedicated newsgroup, respectively.

I am neither a Virtual Treeview nor a Delphi expert and have collected all the answers (with the help of Mike) with quite some effort. In order to avoid the afterwards relatively simple things to become problematic I have written this short guidance. The real problems will appear later.

Have fun with it, Sven.

Preparations

Before we start some preparations are necessary:

- Place a Virtual Treeview component on a form.
- Change the properties as you like.
- A record for node data must be defined.

In order to store the own node data some musing is important. How shall the record look like?

a) All nodes in the tree are equal

In this case a simple record defines the necessary data structure, e.g.:

```
type
  rTreeData = record
  Text: WideString;
  URL: string[255];
  CRC: LongInt;
  isOpened: Boolean;
  ImageIndex: Integer;
```

```
end;
```

b) There are different nodes in the tree (e.g. folders that can have sub nodes)

I will follow this case because my tree will hold folders, which can in turn get own nodes. Since I intent to store created trees in a file in order to restore them later further deliberations are necessary: Suppose a folder node has only a name and a leaf node has a name and a text info field. Potentially, I also want to store a second kind of leaf node, which will for instance have a number instead of the text field. The problem in the context of reading data form a stream is that I must know which data is stored in which order in the stream, because I have to read it in exactly the same order again. Hence I have to determine from the very first information in the stream what information will follow. For instance there is a node name, but then? Is there nothing more or another text information (string) or even an integer value? I think the point is clear. The first data, which I read, has to carry this information.

These deliberations have leaded me to the following solution: I save now in the stream [label]->[name]->[following data]

0 -> 'Folder'

1 -> 'Info node' -> 'Blabla'

2 -> 'Number node' -> 123

I know from the stream I always read an integer value first. Depending whether this is 0, 1 or 2 I have to read - now known - following values. Now let us consider the record.

```
type
  rTreeData = record
  Typ: Integer;
  Name: string[255];
  pNodeData: Pointer;
end;
```

Hey, there is suddenly a pointer in the record. Well, here are some additional comments:

1. Typ is an integer value, from which I can determine what kind of node this is, in my example 1, 2 or 3.
2. Name is the name of the node. This will be needed relatively often because it is also seen as part of the tree and I want to access this information easily (man, I am lazy).
3. The pointer allows (similar to the data property of the tree) a record or even better a class instance to connect.

Now I still have the freedom to define a base class of node. It contains all properties and methods, which all classes will share. And from this I can derive proper sub classes (e.g. text nodes, value nodes etc.). An additional advantage of this record is its fixed size. Hence you can always return the same size in case the tree asks for it (see also property NodeDataSize), but more about that later.

Just one remark: If you don't want to use classes you can also simply define 3 records, which define as first element, a type and which react differently depending on this type.

Alternative solution:

Okay, I admit it. It would of course also be possible to write the type into the stream and read it from the stream separately without saving it as part of the record. The type of the node class is indirectly known because you can ask a class which class name it has (see e.g. class function ClassName) and the class knows it too. So I shall store a node, okay. I pass on the stream to the Node.SaveToFile(Stream) method, which writes, depending on which node class we actually have, automatically the value 1, 2 or 3 into the stream.

During load from stream I read first the value 1, 2 or 3 and decide what class is meant. Then I create an instance of this class and call its LoadFromFile method. Well, this solution is my most preferred and before another one enters my brain I will implement it (Note: in step 5 I will change something).

So I do following:

As you can see from the declaration of the internal node of Virtual Tree

```
TVirtualNode = packed record
  Index, // index of node with regard to its parent
  ChildCount: Cardinal; // number of child nodes
  ...
  ...
  LastChild: PVirtualNode; // link to the node's last child...
  Data: record end; // this is a placeholder, each node gets extra
  // data determined by NodeDataSize
end;
```

there is another record at the end of the record structure. Which exact structure this is will be determined indirectly.

```
type
  rTreeData = record
    Name: string[255]; // the identifier of the node
    ImageIndex: Integer; // the image index of the node
    pNodeData: Pointer;
  end;
```

Let the above record be the structure. The Virtual Treeview does not really know this structure, but it knows how much space must be reserved. We tell it by

```
myVirtualTree.NodeDataSize := SizeOf(rTreeData);
```

Note, even if you want to store only one value, e.g. a pointer as node data, simply return the size, which should be reserved.

Implementation

An empty tree

I begin with an empty tree (no top level nodes are created at design time):

- Either an existing tree is read from a file or
- A top-level node is created.

Before a node can be created you have to determine the size of the actual node data. According to the docs there are three opportunities:

- In the object inspector
- In the OnGetNodeDataSize - event or
- During creation of the form

I decide to use the last variant and will now do the following during form creation:

```
procedure TMyForm.FormCreate(Sender: TObject);

var
  Node: PVirtualNode;

begin
  ...
  // create tree
  MyTree.NodeDataSize := SizeOf(TTreeData);
  if MyForm.filename = '' then begin // if there is no tree to load
    // create tree with top level node
    Node := BookmarkForm.BookmarkTree.AddChild(nil); // adds a top level node
  end
  else
  begin
    // load tree
    ....
  end;
  ....
end;
```

Data for the node

After the call of AddChild data can be assigned. For this a pointer to the self-defined record will be declared and via the function GetNodeData connected with the correct address. By using this pointer we can now access the elements of the record and assign them values.

```
var
  ...
  NodeData: ^rTreeData;

begin
  ...
  // determine data for node
  NodeData := BookmarkForm.BookmarkTree.GetNodeData(Node);
  NodeData.Name := 'new project';
  NodeData.ImageIndex := 0;
  ...
```

Show the node name

The name of the node shall now appear as node identification in the tree. All data about the node as well as the name are unknown to the treeview and it has to query for them.

Every time the identification of the node is needed an event OnGetText will be triggered. In the event handler we return the name of the node in the variable Text. Nothing more is needed.

```
procedure TBookmarkForm.BookmarkTreeGetText(Sender: TBaseVirtualTree;
  Node: PVirtualNode; Column: Integer; TextType: TVSTTextType; var Text: WideString);
var
  NodeData: ^rTreeData;

begin
  NodeData := Sender.GetNodeData(Node);
  // return identifier of the node
  Text := NodeData.Name;
end;
```

The icon for the node

Because I like it colorful I want also to provide an icon for the top-level node. Following steps are necessary to accomplish that:

- A TImageList must be placed onto the form and filled with images
- The property Images of the VirtualTreeview gets assigned this image list
- Implement an OnGetImageIndex event handler.

In the event OnGetImageIndex you can the index be determine which determines in turn which image form the list must be shown.

Because the method is also called for the state icons but I do not want yet to state icons (but I already have assigned and image list to the property StateImages) the value for this case (Kind à ikState) is -1.

```
procedure TBookmarkForm.BookmarkTreeGetImageIndex(Sender: TBaseVirtualTree;
  Node: PVirtualNode; Kind: TVTImageKind; Column: Integer; var Index: Integer);

var
  NodeData: ^rTreeData;

begin
  NodeData := Sender.GetNodeData(Node);
  case Kind of
    ikState: // for the case the state icon has been requested
      Index := -1;
    ikNormal, ikSelected: // normal or the selected icon is required
      Index := NodeData.ImageIndex;
  end;
end;
```

Depending on whether a node is selected or not, different icons shall be shown (see step 6).

Only one node class in the record

Since I want to avoid mixing data in the record and later then data in the node class I decided to change this record

```
type
  TTreeData = record
    Name: string[255]; // the identifier of the node
    ImageIndex: Integer; // the image index of the node
    pNodeData: Pointer;
end;
```

into a record which contains only one pointer to a node class. I declare therefore first a node class

```
TBasicNodeData = class
  ...
end;
```

and then a structure of the form:

```
rTreeData = record
  BasicND: TBasicNodeData;
end;
```

This record always needs 4 bytes for the pointer to the class.

Particular attention is to direct to the event OnGetText. This event will already be called during creation of the node with Tree.AddChild(nil) in order to determine the space the new node's caption will need (but only if no columns were created). At this point however the node class could not yet be initialised (no constructor call yet). Hence for this case

```
if NodeD.BasicND = nil then
```

```
  Text := ''
```

must be returned or you wrap the entire initialization into a BeginUpdate/EndUpdate block and initialized the nodes before EndUpdate is called (e.g. by ValidateNode(Node)).*

Without this provision an access violation would be the result.

Example class declaration

```
unit TreeData;

interface

//=========================================

type
  // declare common node class
  TBasicNodeData = class
  protected
    cName: ShortString;
    cImageIndex: Integer;
  public
    constructor Create; overload;
    constructor Create(vName: ShortString; vIIndex: Integer = 0); overload;

    property Name: ShortString read cName write cName;
    property ImageIndex: Integer read cImageIndex write cImageIndex;
  end;

  // declare new structure for node data
  rTreeData = record
    BasicND: TBasicNodeData;
  end;

implementation

constructor TBasicNodeData.Create;
begin
   { not necessary
   cName := '';
   cImageIndex := 0;
   }
end;

constructor TBasicNodeData.Create(vName: ShortString; vIIndex: Integer = 0);
begin
  cName := vName;
  cImageIndex := vIIndex;
end;

end.
```

Example creation of the tree

```
// Tree will be created when the form is created.
procedure TMyForm.FormCreate(Sender: TObject);

var
   Node: PVirtualNode;
   NodeD: ^rTreeData;

begin
```

```
    ....
    // create tree
    MyTree.NodeDataSize := SizeOf(rTreeData);
    if MainControlForm.filename = '' then
    begin
      // create tree with top level node
      Node := MyTree.AddChild(nil); // adds a node to the root of the tree
      // assign data for this node
      NodeD := MyTree.GetNodeData(Node);
      NodeD.BasicND := TBasicNodeData.Create('new project');
    end
    else
    begin
      // load tree
    end;
    ...
  end;

  // returns the text (the identification) of the node
  procedure TMyForm.MyTreeGetText(Sender: TBaseVirtualTree; Node: PVirtualNode; Column: Integer;
    TextType: TVSTTextType; var Text: WideString);

  var
     NodeD: ^rTreeData;

  begin
    NodeD := Sender.GetNodeData(Node);

    // return the identifier of the node
    if NodeD.BasicND = nil then
      Text := ''
    else
      Text := NodeD.BasicND.Name;
  end;

  // returns the index for image display
  procedure TMyForm.MyTreeGetImageIndex(Sender: TBaseVirtualTree;
     Node: PVirtualNode; Kind: TVTImageKind; Column: Integer; var Index: Integer);

  var
     NodeD: ^rTreeData;

  begin
    NodeD := Sender.GetNodeData(Node);

    case Kind of
      ikState: // for the case the state index has been requested
        Index := -1;
      ikNormal, ikSelected: // normal icon case
        Index := NodeD.BasicND.ImageIndex;
     end;
  end;
```

Icons for selected nodes

If a node is selected a different symbol shall be shown. Therefore I implement a new method

```
function GetImageIndex(focus: Boolean): Integer; virtual;
```

which gets the normal image index or the index for focused nodes depending on whether the node has the focus or not.

Call:

```
Index := NodeD.BasicND.GetImageIndex(Node = Sender.FocusedNode);
```

Implementation of the method:

```
function TBasicNodeData.GetImageIndex(focus: Boolean): Integer;

begin
  if focus then
    Result := cImageIndexFocus
  else
    Result := cImageIndex;
end;
```

where cImageIndex has always the normal index and cImageIndex Focus the index for focused nodes. I assume in this case that the selected index is always one more than the normal index. To ensure this, the constructor is changed this way:

```
constructor TBasicNodeData.Create(vName: ShortString; vIIndex: Integer = 0);

begin
  cName := vName;
  cImageIndex := vIIndex;
  cImageIndexFocus := vIIndex + 1;
end;
```

Adding and deleting nodes

In order to implement and test more functions I want finally an opportunity to create the tree. By using a context menu is shall be possible to add and remove nodes.

Hence I define a popup menu with two entries: [Add] and [Remove]. To have the clicked node getting the focus the option voRightClickSelect must be set to True.

So if Add has been chosen a child node will be created for the focused node:

```
procedure TMyForm.addClick (Sender: TObject);

var
  Node: PVirtualNode;
  NodeD: ^rTreeData;

begin
  // Ok, a node must be added.
  Node := MyTree.AddChild(MyTree.FocusedNode); // adds a node as the last child
  // determine data of node
  NodeD := MyTree.GetNodeData(Node);
  NodeD.BasicND := TBasicNodeData.Create('Child');
end;
```

Caution: What must be done if no node has the focus?

-> e.g. insert the new node as child of a top level nodes.

```
  if BookmarkTree.FocusedNode = nil then
  begin
    // insert as child of the first top level node
    Node := BookmarkTree.AddChild(BookmarkTree.RootNode.FirstChild);
    // determine data for node
    NodeD := BookmarkTree.GetNodeData(Node);
    NodeD.BasicND := TFolderNodeData.Create('new folder');
  end
  else
  begin
    // Ok, a new node must be added.
    Node := BookmarkTree.AddChild(BookmarkTree.FocusedNode);
    // determine data of the node
    NodeD := BookmarkTree.GetNodeData(Node);
    NodeD.BasicND := TFolderNodeData.Create('new folder');
  end;
```

If the node with the focus must be deleted the following happens:

```
procedure TMyForm.delClick (Sender: TObject);

begin
  // The focused node should be removed. The top level must not be deleted however.
  if MyTree.FocusedNode = nil then
    MessageDlg('There was no node selected.', mtInformation, [mbOk], 0)
  else
    // Note: RootNode is the internal (hidden) root node and parent of all top
    // level nodes. To determine whether a node is a top level node you also use
    // GetNodeLevel which returns 0 for top level nodes.
  if MyTree.FocusedNode.Parent = MyTree.RootNode then
    MessageDlg('The project node must not be deleted.', mtInformation, [mbOk], 0)
  else
    MyTree.DeleteNode(MyTree.FocusedNode);
end;
```

I want to prevent, however, that the top-level node gets deleted. Hence I check with the comparison MyTree.FocusedNode.Parent = MyTree.RootNode whether the focused node is not a top-level node. Here you have to consider that the property RootNode returns the (hidden) internal root node, which is the common parent of all top-level nodes.

While we are at deleting nodes:

Every data of the record is automatically free as soon as this is required. In this case it is not enough, however, to free the memory, which holds the pointer to the class (object instance), but it is also necessary to free the memory, which is allocated by the class itself. This happens by calling the destructor of the class in the OnFreeNode event:

```
procedure TMyForm.MyTreeFreeNode(Sender: TBaseVirtualTree; Node: PVirtualNode);

begin
  // Free here the node data (Note: type PtreeData = ^rTreeData).
  PTreeData(Sender.GetNodeData(Node)).BasicND.Free;
end;
```

Adding folder and leafs

Now I am ready to add folders to the tree as well as final nodes, which do not have children. For this I derive two new node classes from the base class.

```
TFolderNodeData = class(TBasicNodeData)
TItemNodeData = class(TBasicNodeData)
```

Depending on which kind of node the user wants to create using the context menu I store a particular class in the node record.

```
NodeD.BasicND := TFolderNodeData.Create('new folder');
NodeD.BasicND := TItemNodeData.Create('new node');
```

These classes contain a new property ChildrenAllowed. Based on this property you can now distinct whether the node with the focus may get children (folder) or not (items).

Storing the tree

Now I can finally implement storing the tree. I have already thought a lot about this step. Let us see if this was worthwhile.

Again a quote from Preparations:

I want to store a node, okay. I hand over the stream to the MyNodeClass.SaveToFile method and this method writes depending upon which node class it actually is automatically the value 1, 2 or 3 as a kind of class ID into the stream (alternatively you can use an enumeration type).

During load I read first the value 1, 2 or 3 from the stream and decide based on it which class we deal with. Then I create an instance of this class and call its method LoadFromFile.

Hint:

It would also be possible to store the class name instead of the ID for the class. During read and creation of the class one could use class references and virtual constructors and save so the case-statement as I did in the OnLoadNode event, to decide which class instance must be created (example see Delphi 5, written by Elmar Warken, Addison-Wesley, chapter 4.3.3, page 439).

Before you can read something it must be written first. Hence I will first implement the necessary procedures to store the tree. Since we care ourselves that the identification of the node gets saved the option toSaveCaption can be removed from StringOptions. This way data is not stored twice.

For saving the tree the procedure

```
procedure TBaseVirtualTree.SaveToFile(const FileName: TFileName);
```

is called. Thereby the structure of the tree is automatically stored. In order to save our additional data there is an event OnSaveNode where we can simply store our data into the provided stream.

```
property OnSaveNode: TVTSaveNodeEvent read FOnSaveNode write FOnSaveNode;
```

If OnSaveNode is triggered then the method SaveNode of the particular node class will be called:

```
procedure TMyForm.MyTreeSaveNode(Sender: TBaseVirtualTree; Node: PVirtualNode; Stream: TStream);

begin
  PTreeData(Sender.GetNodeData(Node)).BasicND.SaveToFile(Stream);
end;
```

In the SaveNode method of the class fields like node name, image index etc. are stored in the tree:

```
procedure TBasicNodeData.SaveNode(Stream: TStream);

var
  size: Integer;

begin
  // save type of the node
  Stream.Write(Art, SizeOf(Art));

  // store cName
  Size := Length(cName) + 1; // include terminating #0
  Stream.Write(Size, SizeOf(Size)); // store length of the string
  Stream.Write(PChar(cName)^, Size); // now the string itself

  // store cImageIndex
  Stream.Write(cImageIndex, SizeOf(cImageIndex));

  // store cImageIndexFocus
  Stream.Write(cImageIndexFocus, SizeOf(cImageIndexFocus));
```

```
    // store cChildrenAllowed
    Stream.Write(cChildrenAllowed, SizeOf(cChildrenAllowed));
  end;
```

Now we can the tree we save also load again. This process could look like:

```
  try
    // load tree
    MyTree.LoadFromFile(MainControlForm.Filename);
  except
    on E: Exception do
    begin
      Application.MessageBox(PChar(E.Message), PChar('Error while loading.'), MB_OK);
      MainControlForm.Filename := '';

      // create tree with top level node (since loading failed)
      Node := MyTree.AddChild(nil);
      NodeD := MyTree.GetNodeData(Node);
      NodeD.BasicND := TBasicNodeData.Create('new project');
    end;
  end;
```

By the call of LoadFromFile the event OnLoadNode will be triggered and consequently the method LoadNode:

```
  procedure TBasicNodeData.LoadNode(Stream: TStream);

  var
    Size: Integer;
    StrBuffer: PChar;

  begin
    // load cName
    Stream.Read(Size, SizeOf(Size)); // length of the string

    StrBuffer := AllocMem(Size); // get temporary memory
    Stream.Read(StrBuffer^, Size); // read the string
    cName := StrBuffer;
    FreeMem(StrBuffer);
    // Alternatively you can simply use:
    // SetLength(cName, Size);
    // Stream.Read(PChar(cName)^, Size);

    // load cImageIndex
    Stream.Read(cImageIndex, SizeOf(cImageIndex));

    // load cImageIndexFocus
    Stream.Read(cImageIndexFocus, SizeOf(cImageIndexFocus));

    // load cChildrenAllowed
    Stream.Read(cChildrenAllowed, SizeOf(cChildrenAllowed));
  end;
```

Two columns in the treeview

Now I want to show two columns in the treeview. Therefore I set the new properties of the tree in the object inspector.

By using Header.Columns you can create the desired columns. After that, you only have to set Header.Options.hoVisible to True and the columns will appear in the treeview.

After you have set all necessary options you can give now the text and the icon for the particular column, respectively. This happens in the already existing event handlers OnGetText and OnGetImageIndex where now also the given column index must be taken into account.

```
procedure TMyForm.MyTreeGetText(Sender: TBaseVirtualTree; Node: PVirtualNode;
```

```
  Column: Integer; TextType: TVSTTextType; var Text: WideString);

var
  NodeD: ^rTreeData;

begin
  NodeD := Sender.GetNodeData(Node);

  // return the the identifier of the node
  if NodeD.BasicND = nil then
    Text := ''
  else
  begin
    case Column of
      -1,
      0: // main column, -1 if columns are hidden, 0 if they are shown
        Text := NodeD.BasicND.Name;
      1:
        Text := 'This text appears in column 2.';
    end;
  end;
end;

procedure TMyForm.MyTreeGetImageIndex(Sender: TBaseVirtualTree; Node: PVirtualNode;
  Kind: TVTImageKind; Column: Integer; var Index: Integer);

var
  NodeD: ^rTreeData;

begin
  NodeD := Sender.GetNodeData(Node);

  if Column = 0 then // icons only in the first column
    case Kind of
      ikState:
        Index := -1;
      ikNormal, ikSelected:
        Index := NodeD.BasicND.GetImageIndex(Node = Sender.FocusedNode);
      ikOverlay: // e.g. to mark a node whose content changed,
        // Note: don't forget to call ImageList.Overlay for the image.
        if NodeD.BasicND.ImageIndex = 4 then
          Index := 6;
    end;
end;
```

Accessing the columns

I want to demonstrate the access to the columns of a TVirtualStringTrees based on an example. In order to store global options, as in Point 2.12 I want to know the width of a column. This information is updated every time an OnColumnResize event is triggered:

```
procedure TBookmarkForm.BookmarkTreeColumnResize(Sender: TBaseVirtualTree; Column: Integer);

var
  NodeD: PTreeData;

begin
  NodeD := Sender.GetNodeData(Sender.RootNode.FirstChild);

  // Keep the new size of the column in the project node.
  TProjectNodeData(NodeD.BasicND).SetHColumnsWidth(
    TVirtualStringTree(Sender).Header.Columns.Items[Column].Width,Column);
end;
```

The exciting part is the type casting of the sender object. In TBaseVirtualTree(see TBaseVirtualTree Class, page 88) the header property is protected and only after conversion (casting) to TVirtualTree it becomes accessible.

Global tree options

Global options like the sizes of the columns, which are adjusted in the project, will be stored as properties of the top-level node. It contains so all project related options.

In order to avoid that all derived classes inherit these fields the top-level node class will be build from a new project node class, which will be derived from the base node class.

The new hierarchy looks now so:

» Base node class... unites the properties of all nodes

» Project node class... enriches the base with management of project related options

» Folder node classes... enriches the base with default properties for all leaf nodes

» Leaf node class... the actual node class (special properties)

Since this involves already very application specific program details I want only make some notes.

The base node class has the ability to store node data. These methods must be declared as virtual and will be overridden in the project node class to allow saving the project data.

Well, now I am ready to work with VirtualTreeview. It will become interesting later again when I will try to drag data from other applications to the tree. But this is a different story...

7 A little code repository by John Knipper

This is just written by me John Knipper.

Description

Don't bother Mike if something is wrong here. I am not related to Mikes company in any way. I'm just doing that because I believe so much in his component, that I would not give you the possibility to miss the opportunity to use it. You won't regret it. I'm not going to enumerate all the nice advantage it has on it's competitors. Because it has so many. The biggest I see is the speed improvement, the multi columns, the automatic allocation of node data and so many more.

You will see that the strong points of the Virtual tree view are not obvious. But you can believe me, this is the best Treeview ever. You will be kinda lost at the beginning, but it's only a matter of forgetting what you know about trees. This is the right way to do it. You will ask yourself why it has not be done like that at the beginning.

Q: How to initially fill the tree?

A: The only information VT needs at startup is the number of root nodes. All other information is queried from the application when they are needed (text, node height etc.). Hence all to do is to set property RootNodeCount to the number of nodes required.

E:

```
VirtualStringTree1.RootNodeCount := 5; // is adding 5 nodes at the root of your tree
```

To initialize the nodes, use the OnInitNode event

Q: How to add a node to the tree?

A: The technique is very similar to the one you used with the standard tree view. The only difference is that you fill the node's data after the insertion of the node

E:

```
var
  Node: PVirtualNode;
  Node := VirtualStringTree1.AddChild(nil); // Adds a node to the root of the Tree.
  Node := VirtualStringTree1.AddChild(ParentNode); // Adds a node as the last child of the given node.
  Node := VirtualStringTree1.InsertNode(Node, amInsertBefore); // Inserts a node as sibling just
before the given node.
```

Alternatively you can use the OnInitChildren event. This event is used when a node is marked as having child nodes and these child nodes are somehow about to be accessed (like iteration, expanding, display etc.).

Q: Where is gone all the information about my node, like text for example ?

A: The text property is gone. You don't need it anymore. The basic idea behind Virtual Treeview is to leave all data management to the application which knows much better how to do this than the tree (see also Related Topics). Every node knows which is its parent and which are their children. Information like the text property, the new hint property, the ImageIndex property and everything else should be stored in the node's data. The tree will ask for it on demand, e.g.

when it needs to show a certain node etc.

E:

```
TTreeData = record
  Text: WideString;
  URL: String[255];
  CRC: LongInt;
  isOpened: Boolean;
  ImageIndex: Integer;
end;
PTreeData = ^TTreeData; // This is a node example.
```

Q: When should I allocate memory for the node data?

A: Never, the VT does it for you. The only thing you have to do is to tell the VT how much memory you need for your node data.

E:

```
VirtualStringTree1.NodeDataSize := SizeOf(TTreeData); // The nodes are reinitialized when you change
that value.
```

If you know how much memory it will take, you can use the NodeDataSize property of the VT and initialize it directly at design time.

Q: When should I fill my nodes data?

A: The ideal place for this is the OnInitNode event.

E:

```
procedure TMainForm.VTInitNode(Sender: TBaseVirtualTree; ParentNode, Node: PVirtualNode; var
InitialStates: TVirtualNodeInitStates);

var
  Level: Integer;
  Data,
  ParentData: PMyNodeData;
  Count: Integer;

begin
  with Sender do
  begin
    Data := GetNodeData(Node);
    ParentData := GetNodeData(ParentNode);
    if Assigned(ParentData) then Level := ParentData.Level + 1
                            else Level := 0;

    case FFillMode of
      0: // fill tree with a specific amount of nodes and levels
        begin
          // determine new node level
          if Level < (LevelsUpDown.Position - 1) then Include(InitialStates, ivsHasChildren);
        end;
      1: // fill tree with one million root nodes (nothing special to do here)
        ;
      2: // fill tree with a certain amount of root nodes (they always get fixed text to test sorting)
        begin
          Data.FixedText := True;
          Data.NewText := Format('Node: %d', [Node.Index]);
        end;
      3: // fill tree with a certain amount of root nodes and a random amount of child nodes
         // up to an absolute amount of ~1 million nodes and with at most 10 levels
        begin
          if Assigned(ParentNode) then Count := ParentNode.ChildCount
```

```
                                    else Count := TVirtualStringTree(Sender).RootNodeCount;
          if (Level < 15) and
             (Random(Count) < (Count div 2)) and
             (FCurrentCount < 1000000) then Include(InitialStates, ivsHasChildren);
        end;
    end;

    Data.Level := Level;
    Node.CheckType := ctTriStateCheckBox;
    case Level of
      1:
        if Random(5) < 2 then Include(InitialStates, ivsDisabled);
    end;
  end;
end;
```

Q: How do I access a node's data?

A: Use GetNodeData(Node) to get a pointer on your nodes data

E: Either use

```
with PTreeData(VirtualStringTree1.GetNodeData(Node))^ do
begin
  Text:= ChangeFileExt(ExtractFileName(FileName), '');
  ImageIndex:= 1; //it's an example ;)
end;
```

Or in that case you can use

```
var
  NodeData: PTreeData;

begin
  NodeData := VirtualStringTree1.GetNodeData(Node);
  NodeData.Text := 'a test';
  NodeData.ImageIndex := 1;
  ...
```

Q: What else can I do with that nodes data pointer?

A: Usually you already have all data in your own structure (database, file etc.) so you need only to supply an identifier or a pointer into your own structure. This prevents your application from doubling the data just for display which in turn saves a remarkable amount of memory.

E: You could connect a TBookmark to the data. To display the name of your customer in a VT :

```
procedure TFRM_WWW_main.vFavTreeGetText(Sender: TBaseVirtualTree; Node: PVirtualNode; Column: Integer;
TextType: TVSTTextType; var Text: WideString);
begin
  // Column is -1 if the header is hidden or no columns are defined
  if Column < 0 then Exit;
  if TVirtualStringTree(Sender).Header.Columns[Column].Text = 'Customer Name' then
  begin
    Table.GotoBookmark(TBookmark(Sender.GetNodeData(Node)));
    Text := Table.FieldByName('Name').asString;
  end;
end;
```

Q: A move of a scrollbar's thumb doesn't directly scroll the tree. What to do?

A:

```
VirtualStringTree1.VertScrollBar.Track := True;
```

Q: How can I display text for other columns?

A: In the OnGetText event, check the column index.

E:

```
procedure TFRM_WWW_main.vFavTreeGetText(Sender: TBaseVirtualTree; Node: PVirtualNode; Column: Integer;
TextType: TVSTTextType; var Text: WideString);
begin
  case Column of
    -1, // main column, -1 if columns are hidden, 0 if they are shown
    0:
      Text := 'Text of column 1';
    1:
      Text := 'Text of column 2';
    2:
      Text := 'Text of column 3';
  end;
end;
```

Q: When do I tell which icon to use?

A: It's the same principle as for the OnGetText event. With the exception that you must tell which icon to use in 3 cases: the normal icon, the selected icon and the state icon.

E:

```
procedure TFRM_WWW_main.vFavTreeGetImageIndex(Sender: TBaseVirtualTree; Node: PVirtualNode; Kind:
TVTImageKind; Column: Integer; var Index: Integer);
begin
  if Kind = ikState then
  begin
    Index := 2;
  end
  else
    if (Kind = ikNormal) or (Kind = ikSelected) then
    begin
      Index := 1;
    end;
end;
```

or just use

```
procedure TFRM_WWW_main.vFavTreeGetImageIndex(Sender: TBaseVirtualTree; Node: PVirtualNode; Kind:
TVTImageKind; Column: Integer; var Index: Integer);
begin
  case Kind of
    ikState:
      Index := 2;
    ikNormal,
    ikSelected:
      Index := 1;
  end;
end;
```

Ok, here we are. This is only a small introduction to help you begin with Virtual Treeview. There are many more useful functions. Nearly everything was done for you. Thank you very much for your work Mike.

8 Questions and Answers

Got some basic questions and need an answer - look here:

Q: How to initially fill the tree?

A: The only information VT needs at startup is the number of root nodes.

Description

All other information is queried from the application when they are needed (text, child count etc.). Hence all to do is to set property RootNodeCount to the number of nodes required.

Q: When I change the text of a node in code then often the display is not updated. What must I do to make selection etc. working again?

A: The Virtual String Tree class keeps the caption's width for each node to allow quick hit tests. But since the captions are not stored in the tree they might get out of synch with the cached width. So if you change a node's text or only its width somehow (e.g. making it bold in OnPaintText) then you have to tell the tree about this event. You can do this by calling InvalidateNode. For changes in an event, though, you should not call InvaldiateNode all the time but rather store the text attributes somewhere and force recalculation only once.

Q: Why doesn't the horizontal scroll bar stay constant while scrolling vertically and columns are unused?

A: VT holds (except a few important things for the overall structure) no information about a node to save memory and provide high speed access. This implies, though, that it only knows the width of the items currently displayed in the client area. Hence the horizontal scroll bar reflects only the width of the largest node currently in view. When columns are used then the width is determined by the overall width of the header.

Q: Why is the horizontal scroll bar not updated when scrolling vertically using the scroll thumb?

A: To avoid unnecessary flickering and to keep high speed response the horizontal scroll bar is updated after the scroll thumb has been released. You cannot scroll horizontally while scrolling vertically, so the horizontal scroll bar doesn't need to be updated while thumb tracking. When columns are used then the width is constant anyway and the horizontal scrollbar does not need an update.

Q: How to assign and access my own data to/on a node?

A: VT does not hide any information about the internal structure of the node from the application. And the best place to hold data specific to a node is the node itself. So there's a user definable area at the end of each node record which can be used to store application data. Usually you already have all data in your own structure (database, file etc.) so you need only to supply an identifier or a pointer (link) into your own structure. This prevents your application from doubling the data just for display which in turn can save a remarkable amount of memory. As the space requirements may vary from application to application the amount of user data space can be globally adjusted by the property NodeDataSize. In order to avoid ugly pointer math there's a function GetNodeData which returns a pointer which directly corresponds to the user data area (it points to the first byte in that area). I strongly recommend to use GetNodeData only (instead directly accessing a node's data area) because specific tree classes may additionally allocate data in the user data area and these parts needs to be taken into account. Assign the returned pointer to your own variable of the correct type (or just cast the pointer) and access your own data as usual. Note: Setting NodeDataSize will clear the entire tree and build it from scratch using this new size as all node records have to be reallocated.

Q: Do I need to check if a node's data is successfully allocated?

A: No, user data is allocated with the node (actually it is part of a node) so the memory allocation function takes care of initialization.

Q: How to get the currently focused node and the target node during a drag'n drop operation?

A: Just query property FocusedNode and DropTargetNode, respectively.

Q: When to free my own node specific data?

A: Use OnFreeNode as central routine to release/disconnect all your data (just like as you should use OnNodeInit to allocate/attach your data to the node).

Q: How can I know which node am I working on?

A: You might want to access the currently FocusedNode to add child nodes to etc. or you might want to use the drop target to act on during a drag'n drop operation etc. But usually you are working on the selection. You have two opportunities to get a list of currently selected nodes. One is the GetFirstSelected/GetNextSelected pair which is really fast but returns the nodes precisely as they are in the internal selection array (which is ordered by memory locations, not logically). Or you can use GetSortedSelection which fills a dynamic array with node references in logical (structural) order.

Q: Is user data saved while doing drag'n drop or saving/restoring nodes?

A: This question implies another question, which I want to answer first: Yes, the same mechanism to save and load nodes is used for drag'n drop as for streaming to/from a file. Because of potentially large node data and/or many nodes the user data is not saved by default with a node. There are the OnSaveNode and OnLoadNode events which provide the application with a stream to store its node data in.

Q: Where should I update my external resources (like a database) involved by any node manipulation?

A: There are several events which could be used. First there is the set of edit events (OnEditCancelled, OnEditing, TBaseVirtualTree(see TBaseVirtualTree Class, page 88)) which indicate the cancellation, start and successful finish of an edit event, respectively. These events are used to generally indicate editing of a node. Especially for the node's text in a TVirtualStringTree(see TVirtualStringTree Class, page 402) another event might ease your life. It's the OnNewText event. This is a good place to set a record's description/caption in a database etc.

When it comes to structure changes then usually much more work is involved to keep external data in sync. For general notifications of such a change you might want to use OnStructureChange. This event might often be enough, in particular when also OnInitNode, OnInitChildren and OnFreeNode are considered. But for cut, copy and paste as well as drag'n drop even more care must be taken, since a node might move within the tree what then involves a move of a database record or a file etc. For this kind of action the event pairs OnNodeCopying/OnNodeCopied and OnNodeMoving/TBaseVirtualTree(see TBaseVirtualTree Class, page 88) have been introduced. As with all those pairs you can reject copying or moving a node.

Note: These events do only appear for the top node which represents a sub tree! For example if the user drags the second and the third top level node of a tree to a Word document then you'll get only two events, one for each selected node, but not for any child node even if they are selected too. You can still walk through the child nodes if you need to by using e.g. IterateSubTree, but usually a tree represents a hierarchical structure which is recursively defined which avoids the need to update each and every of probably many child nodes.

9 Licensing

Virtual Treeview License Agreement

The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License.

Description

You may obtain a copy of the License at www.mozilla.org/MPL.

Alternatively, you may redistribute this library, use and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. You may obtain a copy of the LGPL at www.gnu.org/copyleft.

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The original code is VirtualTrees.pas, released September 30, 2000.

The initial developer of the original code is digital publishing AG (www.digitalpublishing.de).

Virtual Treeview is written, published and maintaned by

Mike Lischke (public@soft-gems.net, www.soft-gems.net).

10 Virtual Treeview

EVirtualTreeError(see EVirtualTreeError Class, page 87)

TBaseVirtualTree(see TBaseVirtualTree Class, page 88)

TBaseVirtualTree is the main and base class for all other Virtual Treeview descendants.

TBufferedString(see TBufferedString Class, page 241)

TClipboardFormatList(see TClipboardFormatList Class, page 243)

Not documented.

TClipboardFormats(see TClipboardFormats Class, page 246)

List of strings describing clipboard formats.

TCriticalSection(see TCriticalSection Class, page 248)

Not documented.

TCustomStringTreeOptions(see TCustomStringTreeOptions Class, page 250)

Enhanced options class for string trees.

TCustomVirtualDrawTree(see TCustomVirtualDrawTree Class, page 252)

Simple owner draw descendant of the base tree.

TCustomVirtualStringTree(see TCustomVirtualStringTree Class, page 274)

Descendant of TBaseVirtualTree(see TBaseVirtualTree Class, page 88), which is able to manage node captions on its own

TCustomVirtualTreeOptions(see TCustomVirtualTreeOptions Class, page 308)

Organizes all tree options into subproperties for easier managment.

TEnumFormatEtc(see TEnumFormatEtc Class, page 311)

TScrollBarOptions(see TScrollBarOptions Class, page 313)

TStringEditLink(see TStringEditLink Class, page 316)

TStringEditLink is the standard node editor of a TVirtualStringTree(see TVirtualStringTree Class, page 402).

TStringTreeOptions(see TStringTreeOptions Class, page 320)

Options class used in the string tree and its descentants.

TVirtualDrawTree(see TVirtualDrawTree Class, page 323)

Descendant of TBaseVirtualTree(see TBaseVirtualTree Class, page 88), which passes node paint events through to the application (similar to a draw grid)

TVirtualStringTree(see TVirtualStringTree Class, page 402)

Descentant of TBaseVirtualTree(see TBaseVirtualTree Class, page 88) which is able to manage node captions on its own.

TVirtualTreeColumn(see TVirtualTreeColumn Class, page 485)

Represents a column in a Virtual Treeview.

TVirtualTreeColumns(see TVirtualTreeColumns Class, page 497)

Collection class, which holds the columns for the tree.

TVirtualTreeHintWindow(see TVirtualTreeHintWindow Class, page 510)

Internally used hint window class to support Unicode hints.

TVirtualTreeOptions(see TVirtualTreeOptions Class, page 513)

Collects all binary options of the tree control into one place for easier access.

TVTColors(see TVTColors Class, page 515)

Collects all color related options for the tree control.

TVTDataObject(see TVTDataObject Class, page 521)

Implementation of an IDataObject interface.

TVTDragImage(see TVTDragImage Class, page 529)

Not documented.

TVTDragManager(see TVTDragManager Class, page 536)

Not documented.

TVTEdit(see TVTEdit Class, page 539)

Not documented.

TVTHeader(see TVTHeader Class, page 544)
Not documented.

TVTHeaderPopupMenu(see TVTHeaderPopupMenu Class, page 557)
Not documented.

TWideBufferedString(see TWideBufferedString Class, page 560)
Not documented.

TWorkerThread(see TWorkerThread Class, page 561)
Not documented.

TWriterHack(see TWriterHack Class, page 564)
Not documented.

Constants

AlignmentToDrawFlag(see AlignmentToDrawFlag Constant, page 672)
Not documented.

AllocIncrement(see AllocIncrement Constant, page 672)
Not documented.

BaseChunk(see BaseChunk Constant, page 672)
Not documented.

CacheThreshold(see CacheThreshold Constant, page 673)
Number of nodes a tree must at least have to start caching and at the same time the maximum number of nodes between two cache entries.

CaptionChunk(see CaptionChunk Constant, page 673)
Not documented.

CFSTR_CSV(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

CFSTR_HTML(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

CFSTR_RTF(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

CFSTR_RTFNOOBJS(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

CFSTR_VIRTUALTREE(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

CFSTR_VTREFERENCE(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

ChangeTimer(see ChangeTimer Constant, page 674)
Not documented.

Check button image indices(see page 674)
ckButtonDisabled(see Check button image indices, page 674)
ckButtonHot(see Check button image indices, page 674)
ckButtonNormal(see Check button image indices, page 674)
ckButtonPressed(see Check button image indices, page 674)
ckCheckCheckedDisabled(see Check button image indices, page 674)
ckCheckCheckedHot(see Check button image indices, page 674)
ckCheckCheckedNormal(see Check button image indices, page 674)
ckCheckCheckedPressed(see Check button image indices, page 674)
ckCheckMixedDisabled(see Check button image indices, page 674)
ckCheckMixedHot(see Check button image indices, page 674)
ckCheckMixedNormal(see Check button image indices, page 674)
ckCheckMixedPressed(see Check button image indices, page 674)
ckCheckUncheckedDisabled(see Check button image indices, page 674)
ckCheckUncheckedHot(see Check button image indices, page 674)

ckCheckUncheckedNormal(see Check button image indices, page 674)
ckCheckUncheckedPressed(see Check button image indices, page 674)
ckEmpty(see Check button image indices, page 674)
ckRadioCheckedDisabled(see Check button image indices, page 674)
ckRadioCheckedHot(see Check button image indices, page 674)
ckRadioCheckedNormal(see Check button image indices, page 674)
ckRadioCheckedPressed(see Check button image indices, page 674)
ckRadioUncheckedDisabled(see Check button image indices, page 674)
ckRadioUncheckedHot(see Check button image indices, page 674)
ckRadioUncheckedNormal(see Check button image indices, page 674)
ckRadioUncheckedPressed(see Check button image indices, page 674)
ClipboardStates(see ClipboardStates Constant, page 675)
Not documented.

CLSID_DragDropHelper(see CLSID_DragDropHelper Constant, page 675)
Not documented.

CM_AUTOADJUST(see CM_AUTOADJUST Constant, page 675)
Not documented.

CM_DENYSUBCLASSING(see CM_DENYSUBCLASSING Constant, page 676)
Not documented.

Copyright(see Copyright Constant, page 676)
Not documented.

crHeaderSplit(see crHeaderSplit Constant, page 676)
Not documented.

DefaultAnimationOptions(see DefaultAnimationOptions Constant, page 677)
Not documented.

DefaultAutoOptions(see DefaultAutoOptions Constant, page 677)
Not documented.

DefaultColumnOptions(see DefaultColumnOptions Constant, page 677)
Not documented.

DefaultMiscOptions(see DefaultMiscOptions Constant, page 678)
Not documented.

DefaultPaintOptions(see DefaultPaintOptions Constant, page 678)
Not documented.

DefaultScrollUpdateFlags(see DefaultScrollUpdateFlags Constant, page 678)
Not documented.

DefaultSelectionOptions(see DefaultSelectionOptions Constant, page 679)
Not documented.

DefaultStringOptions(see DefaultStringOptions Constant, page 679)
Not documented.

EditTimer(see EditTimer Constant, page 679)
Not documented.

ExpandTimer(see ExpandTimer Constant, page 680)
Not documented.

FadeAnimationStepCount(see FadeAnimationStepCount Constant, page 680)
Not documented.

Grays(see Grays Constant, page 680)
Not documented.

hcTFCannotSetUserData(see hcTFCannotSetUserData Constant, page 681)
Not documented.

hcTFClipboardFailed(see hcTFClipboardFailed Constant, page 681)
Not documented.

hcTFCorruptStream1(see hcTFCorruptStream1 Constant, page 681)

Not documented.

hcTFCorruptStream2(see hcTFCorruptStream2 Constant, page 682)
Not documented.

hcTFEditLinkIsNil(see hcTFEditLinkIsNil Constant, page 682)
Not documented.

hcTFStreamTooSmall(see hcTFStreamTooSmall Constant, page 682)
Not documented.

hcTFWrongMoveError(see hcTFWrongMoveError Constant, page 683)
Not documented.

hcTFWrongStreamFormat(see hcTFWrongStreamFormat Constant, page 683)
Not documented.

hcTFWrongStreamVersion(see hcTFWrongStreamVersion Constant, page 683)
Not documented.

HeaderTimer(see HeaderTimer Constant, page 684)
Not documented.

IID_IDragSourceHelper(see IID_IDragSourceHelper Constant, page 684)
Not documented.

IID_IDropTarget(see IID_IDropTarget Constant, page 684)
Not documented.

IID_IDropTargetHelper(see IID_IDropTargetHelper Constant, page 685)
Not documented.

InvalidColumn(see InvalidColumn Constant, page 685)
Not documented.

MagicID(see MagicID Constant, page 685)
Not documented.

MinimumTimerInterval(see MinimumTimerInterval Constant, page 686)
Not documented.

MouseButtonDown(see MouseButtonDown Constant, page 686)
Not documented.

NoColumn(see NoColumn Constant, page 686)
Not documented.

NodeChunk(see NodeChunk Constant, page 687)
Not documented.

OptionMap(see OptionMap Constant, page 687)
Not documented.

PressedState(see PressedState Constant, page 687)
Not documented.

RTLFlag(see RTLFlag Constant, page 688)
Not documented.

SCannotSetUserData(see SCannotSetUserData Constant, page 688)
Not documented.

SClipboardFailed(see SClipboardFailed Constant, page 688)
Not documented.

SCorruptStream1(see SCorruptStream1 Constant, page 689)
Not documented.

SCorruptStream2(see SCorruptStream2 Constant, page 689)
Not documented.

ScrollTimer(see ScrollTimer Constant, page 689)
Not documented.

SearchTimer(see SearchTimer Constant, page 690)
Not documented.

SEditLinkIsNil(see SEditLinkIsNil Constant, page 690)
Not documented.

ShadowSize(see ShadowSize Constant, page 690)
Size in pixels of the hint shadow.

SID_IDragSourceHelper(see SID_IDragSourceHelper Constant, page 691)
Not documented.

SID_IDropTarget(see SID_IDropTarget Constant, page 691)
Not documented.

SID_IDropTargetHelper(see SID_IDropTargetHelper Constant, page 691)
Not documented.

SStreamTooSmall(see SStreamTooSmall Constant, page 692)
Not documented.

StructureChangeTimer(see StructureChangeTimer Constant, page 692)
Not documented.

SWrongMoveError(see SWrongMoveError Constant, page 692)
Not documented.

SWrongStreamFormat(see SWrongStreamFormat Constant, page 693)
Not documented.

SWrongStreamVersion(see SWrongStreamVersion Constant, page 693)
Not documented.

SysGrays(see SysGrays Constant, page 693)
Not documented.

TreeNodeSize(see TreeNodeSize Constant, page 694)
Not documented.

UnpressedState(see UnpressedState Constant, page 694)
Not documented.

UserChunk(see UserChunk Constant, page 694)
Not documented.

UtilityImageSize(see UtilityImageSize Constant, page 695)
Not documented.

VTHeaderStreamVersion(see VTHeaderStreamVersion Constant, page 695)
Not documented.

VTTreeStreamVersion(see VTTreeStreamVersion Constant, page 695)
Not documented.

VTVersion(see VTVersion Constant, page 696)
Not documented.

WideCR(see WideCR Constant, page 696)
Not documented.

WideLF(see WideLF Constant, page 696)
Not documented.

WideLineSeparator(see WideLineSeparator Constant, page 697)
Not documented.

WideNull(see WideNull Constant, page 697)
Not documented.

WM_CHANGESTATE(see WM_CHANGESTATE Constant, page 697)
Not documented.

XPDarkGradientColor(see XPDarkGradientColor Constant, page 698)
Not documented.

XPDarkSplitBarColor(see XPDarkSplitBarColor Constant, page 698)
Not documented.

XPDownInnerLineColor(see XPDownInnerLineColor Constant, page 698)

Not documented.

XPDownMiddleLineColor(see XPDownMiddleLineColor Constant, page 699)
Not documented.

XPDownOuterLineColor(see XPDownOuterLineColor Constant, page 699)
Not documented.

XPLightSplitBarColor(see XPLightSplitBarColor Constant, page 699)
Not documented.

XPMainHeaderColorDown(see XPMainHeaderColorDown Constant, page 700)
Not documented.

XPMainHeaderColorHover(see XPMainHeaderColorHover Constant, page 700)
Not documented.

XPMainHeaderColorUp(see XPMainHeaderColorUp Constant, page 700)
Not documented.

Functions

AlphaBlend(see AlphaBlend Function, page 565)
General purpose procedure to blend one bitmap to another.

DrawTextW(see DrawTextW Function, page 566)
Paint support procedure.

EnumerateVTClipboardFormats(see EnumerateVTClipboardFormats Function, page 567)
Not documented.

EnumerateVTClipboardFormats(see EnumerateVTClipboardFormats Function, page 567)
Not documented.

GetVTClipboardFormatDescription(see GetVTClipboardFormatDescription Function, page 567)
Not documented.

PrtStretchDrawDIB(see PrtStretchDrawDIB Function, page 568)
Not documented.

RegisterVTClipboardFormat(see RegisterVTClipboardFormat Function, page 568)
Methods to register a certain clipboard format for a given tree class.

RegisterVTClipboardFormat(see RegisterVTClipboardFormat Function, page 568)
Methods to register a certain clipboard format for a given tree class.

ShortenString(see ShortenString Function, page 568)
General purpose routine to shorten a Unicode string to a given maximum size.

TreeFromNode(see TreeFromNode Function, page 569)
General purpose routine to get the tree to which a node belongs.

Structs and Records

TBaseChunk(see TBaseChunk Record, page 571)
Not documented.

TBaseChunkBody(see TBaseChunkBody Record, page 571)
Not documented.

TCacheEntry(see TCacheEntry Record, page 571)
Not documented.

TChunkHeader(see TChunkHeader Record, page 572)
Not documented.

TClipboardFormatEntry(see TClipboardFormatEntry Record, page 572)
Not documented.

TClipboardFormatListEntry(see TClipboardFormatListEntry Record, page 573)
Not documented.

THeaderPaintInfo(see THeaderPaintInfo Record, page 573)
Not documented.

THitInfo(see THitInfo Record, page 574)

Not documented.

TInternalStgMedium(see TInternalStgMedium Record, page 574)
Not documented.

TRealWMNCPaint(see TRealWMNCPaint Record, page 574)
Not documented.

TSHDragImage(see TSHDragImage Record, page 575)
Not documented.

TToggleAnimationData(see TToggleAnimationData Record, page 575)
Not documented.

TVirtualNode(see TVirtualNode Record, page 576)
Not documented.

TVTHintData(see TVTHintData Record, page 577)
Not documented.

TVTImageInfo(see TVTImageInfo Record, page 577)
Not documented.

TVTPaintInfo(see TVTPaintInfo Record, page 578)
Not documented.

TVTReference(see TVTReference Record, page 578)
Not documented.

TVTTooltipLineBreakStyle(see TVTTooltipLineBreakStyle Enumeration, page 579)
Not documented.

TWMPrint(see TWMPrint Record, page 579)
Not documented.

Topics

Classes(see page 86)
These are all classes that are contained in this documentation.

Functions(see page 565)
These are all functions that are contained in this documentation.

Structs and Records(see page 569)
These are all structs and records that are contained in this documentation.

Types(see page 580)
These are all types that are contained in this documentation.

Variables(see page 656)
These are all variables that are contained in this documentation.

Constants(see page 667)
These are all constants that are contained in this documentation.

Symbol Reference(see page 701)
These are all symbols available in this documentation.

Types

PCardinal(see PCardinal Type, page 587)
Not documented.

PClipboardFormatListEntry(see PClipboardFormatListEntry Type, page 587)
Not documented.

PSHDragImage(see PSHDragImage Type, page 588)
Not documented.

PVirtualNode(see PVirtualNode Type, page 588)
Not documented.

PVTHintData(see PVTHintData Type, page 588)
Not documented.

PVTReference(see PVTReference Type, page 589)

Not documented.

TAddHeaderPopupItemEvent(see TAddHeaderPopupItemEvent Type, page 589)
Not documented.

TAutoScrollInterval(see TAutoScrollInterval Type, page 589)
Not documented.

TCache(see TCache Type, page 590)
Not documented.

TCardinalArray(see TCardinalArray Type, page 590)
Not documented.

TChangeStates(see TChangeStates Type, page 590)
Not documented.

TColumnChangeEvent(see TColumnChangeEvent Type, page 591)
Not documented.

TColumnIndex(see TColumnIndex Type, page 591)
Not documented.

TColumnPosition(see TColumnPosition Type, page 591)
Not documented.

TColumnsArray(see TColumnsArray Type, page 592)
Not documented.

TDragOperations(see TDragOperations Type, page 592)
Not documented.

TFormatArray(see TFormatArray Type, page 592)
Not documented.

TFormatEtcArray(see TFormatEtcArray Type, page 593)
Not documented.

TGetFirstNodeProc(see TGetFirstNodeProc Type, page 593)
Not documented.

TGetNextNodeProc(see TGetNextNodeProc Type, page 593)
Not documented.

THeaderPaintElements(see THeaderPaintElements Type, page 594)
Not documented.

THeaderStates(see THeaderStates Type, page 594)
Not documented.

THitPositions(see THitPositions Type, page 594)
Not documented.

TImageIndex(see TImageIndex Type, page 595)
Not documented.

TIndexArray(see TIndexArray Type, page 595)
Not documented.

TInternalStgMediumArray(see TInternalStgMediumArray Type, page 595)
Not documented.

TLineImage(see TLineImage Type, page 596)
Not documented.

TMagicID(see TMagicID Type, page 596)
Not documented.

TMouseButtons(see TMouseButtons Type, page 596)
Not documented.

TNodeArray(see TNodeArray Type, page 597)
Not documented.

TScrollDirections(see TScrollDirections Type, page 597)
Not documented.

TScrollUpdateOptions(see TScrollUpdateOptions Type, page 597)
Not documented.

TTreeOptionsClass(see TTreeOptionsClass Type, page 598)
Not documented.

TVirtualNodeInitStates(see TVirtualNodeInitStates Type, page 598)
Not documented.

TVirtualNodeStates(see TVirtualNodeStates Type, page 598)
Not documented.

TVirtualTreeClass(see TVirtualTreeClass Type, page 599)
Not documented.

TVirtualTreeColumnClass(see TVirtualTreeColumnClass Type, page 599)
Not documented.

TVirtualTreeColumnsClass(see TVirtualTreeColumnsClass Type, page 599)
Not documented.

TVirtualTreeStates(see TVirtualTreeStates Type, page 600)
Not documented.

TVSTGetTextEvent(see TVSTGetTextEvent Type, page 600)
Not documented.

TVSTNewTextEvent(see TVSTNewTextEvent Type, page 601)
Not documented.

TVSTShortenStringEvent(see TVSTShortenStringEvent Type, page 601)
Not documented.

TVTAdvancedHeaderPaintEvent(see TVTAdvancedHeaderPaintEvent Type, page 601)
Not documented.

TVTAfterCellPaintEvent(see TVTAfterCellPaintEvent Type, page 602)
Not documented.

TVTAfterItemEraseEvent(see TVTAfterItemEraseEvent Type, page 602)
Not documented.

TVTAfterItemPaintEvent(see TVTAfterItemPaintEvent Type, page 602)
Not documented.

TVTAnimationCallback(see TVTAnimationCallback Type, page 603)
Not documented.

TVTAnimationOptions(see TVTAnimationOptions Type, page 603)
Not documented.

TVTAutoOptions(see TVTAutoOptions Type, page 603)
Not documented.

TVTBackgroundPaintEvent(see TVTBackgroundPaintEvent Type, page 604)
Not documented.

TVTBeforeCellPaintEvent(see TVTBeforeCellPaintEvent Type, page 604)
Not documented.

TVTBeforeItemEraseEvent(see TVTBeforeItemEraseEvent Type, page 604)
Not documented.

TVTBeforeItemPaintEvent(see TVTBeforeItemPaintEvent Type, page 605)
Not documented.

TVTBias(see TVTBias Type, page 605)
Not documented.

TVTChangeEvent(see TVTChangeEvent Type, page 605)
Not documented.

TVTChangingEvent(see TVTChangingEvent Type, page 606)
Not documented.

TVTCheckChangingEvent(see TVTCheckChangingEvent Type, page 606)

Not documented.

TVTColumnClickEvent(see TVTColumnClickEvent Type, page 606)
Not documented.

TVTColumnDblClickEvent(see TVTColumnDblClickEvent Type, page 607)
Not documented.

TVTColumnOptions(see TVTColumnOptions Type, page 607)
Not documented.

TVTCompareEvent(see TVTCompareEvent Type, page 607)
Not documented.

TVTCreateDataObjectEvent(see TVTCreateDataObjectEvent Type, page 608)
Not documented.

TVTCreateDragManagerEvent(see TVTCreateDragManagerEvent Type, page 608)
Not documented.

TVTCreateEditorEvent(see TVTCreateEditorEvent Type, page 609)
Not documented.

TVTDragAllowedEvent(see TVTDragAllowedEvent Type, page 609)
Not documented.

TVTDragDropEvent(see TVTDragDropEvent Type, page 609)
Not documented.

TVTDragImageStates(see TVTDragImageStates Type, page 610)
Not documented.

TVTDragOverEvent(see TVTDragOverEvent Type, page 610)
Not documented.

TVTDrawHintEvent(see TVTDrawHintEvent Type, page 610)
Not documented.

TVTDrawNodeEvent(see TVTDrawNodeEvent Type, page 611)
Not documented.

TVTEditCancelEvent(see TVTEditCancelEvent Type, page 611)
Not documented.

TVTEditChangeEvent(see TVTEditChangeEvent Type, page 611)
Not documented.

TVTEditChangingEvent(see TVTEditChangingEvent Type, page 612)
Not documented.

TVTFocusChangeEvent(see TVTFocusChangeEvent Type, page 612)
Not documented.

TVTFocusChangingEvent(see TVTFocusChangingEvent Type, page 612)
Not documented.

TVTFreeNodeEvent(see TVTFreeNodeEvent Type, page 613)
Not documented.

TVTGetCursorEvent(see TVTGetCursorEvent Type, page 613)
Not documented.

TVTGetHeaderCursorEvent(see TVTGetHeaderCursorEvent Type, page 613)
Not documented.

TVTGetHintSizeEvent(see TVTGetHintSizeEvent Type, page 614)
Not documented.

TVTGetImageEvent(see TVTGetImageEvent Type, page 614)
Not documented.

TVTGetLineStyleEvent(see TVTGetLineStyleEvent Type, page 614)
Not documented.

TVTGetNodeDataSizeEvent(see TVTGetNodeDataSizeEvent Type, page 615)
Not documented.

TVTGetNodeProc(see TVTGetNodeProc Type, page 615)
Not documented.

TVTGetNodeWidthEvent(see TVTGetNodeWidthEvent Type, page 615)
Not documented.

TVTGetUserClipboardFormatsEvent(see TVTGetUserClipboardFormatsEvent Type, page 616)
Not documented.

TVTHeaderClass(see TVTHeaderClass Type, page 616)
Not documented.

TVTHeaderClickEvent(see TVTHeaderClickEvent Type, page 616)
Not documented.

TVTHeaderDraggedEvent(see TVTHeaderDraggedEvent Type, page 617)
Not documented.

TVTHeaderDraggedOutEvent(see TVTHeaderDraggedOutEvent Type, page 617)
Not documented.

TVSTGetHintEvent(see TVSTGetHintEvent Type, page 617)
Not documented.

TVTHeaderDraggingEvent(see TVTHeaderDraggingEvent Type, page 618)
Not documented.

TVTHeaderMouseEvent(see TVTHeaderMouseEvent Type, page 618)
Not documented.

TVTHeaderMouseMoveEvent(see TVTHeaderMouseMoveEvent Type, page 618)
Not documented.

TVTHeaderNotifyEvent(see TVTHeaderNotifyEvent Type, page 619)
Not documented.

TVTHeaderOptions(see TVTHeaderOptions Type, page 619)
Not documented.

TVTHeaderPaintEvent(see TVTHeaderPaintEvent Type, page 619)
Not documented.

TVTHeaderPaintQueryElementsEvent(see TVTHeaderPaintQueryElementsEvent Type, page 620)
Not documented.

TVTHeaderPopupOptions(see TVTHeaderPopupOptions Type, page 620)
Not documented.

TVTHelpContextEvent(see TVTHelpContextEvent Type, page 620)
Not documented.

TVTHotNodeChangeEvent(see TVTHotNodeChangeEvent Type, page 621)
Not documented.

TVTIncrementalSearchEvent(see TVTIncrementalSearchEvent Type, page 621)
Not documented.

TVTInitChildrenEvent(see TVTInitChildrenEvent Type, page 621)
Not documented.

TVTInitNodeEvent(see TVTInitNodeEvent Type, page 622)
Not documented.

TVTInternalPaintOptions(see TVTInternalPaintOptions Type, page 622)
Not documented.

TVTKeyActionEvent(see TVTKeyActionEvent Type, page 622)
Not documented.

TVTMeasureItemEvent(see TVTMeasureItemEvent Type, page 623)
Not documented.

TVTMiscOptions(see TVTMiscOptions Type, page 623)
Not documented.

TVTNodeCopiedEvent(see TVTNodeCopiedEvent Type, page 623)

Not documented.

TVTNodeCopyingEvent(see TVTNodeCopyingEvent Type, page 624)
Not documented.

TVTNodeMovedEvent(see TVTNodeMovedEvent Type, page 624)
Not documented.

TVTNodeMovingEvent(see TVTNodeMovingEvent Type, page 624)
Not documented.

TVTPaintEvent(see TVTPaintEvent Type, page 625)
Not documented.

TVTPaintOptions(see TVTPaintOptions Type, page 625)
Not documented.

TVTPaintText(see TVTPaintText Type, page 625)
Not documented.

TVTPopupEvent(see TVTPopupEvent Type, page 626)
Not documented.

TVTRenderOLEDataEvent(see TVTRenderOLEDataEvent Type, page 626)
Not documented.

TVTSaveNodeEvent(see TVTSaveNodeEvent Type, page 627)
Not documented.

TVTScrollEvent(see TVTScrollEvent Type, page 627)
Not documented.

TVTScrollIncrement(see TVTScrollIncrement Type, page 627)
Not documented.

TVTSelectionOptions(see TVTSelectionOptions Type, page 628)
Not documented.

TVTStateChangeEvent(see TVTStateChangeEvent Type, page 628)
Not documented.

TVTStringOptions(see TVTStringOptions Type, page 628)
Not documented.

TVTStructureChangeEvent(see TVTStructureChangeEvent Type, page 629)
Not documented.

TVTTransparency(see TVTTransparency Type, page 629)
Not documented.

TVTUpdatingEvent(see TVTUpdatingEvent Type, page 629)
Not documented.

TWMContextMenu(see TWMContextMenu Type, page 630)
Not documented.

TWMPrintClient(see TWMPrintClient Type, page 630)
Not documented.

TAddPopupItemType(see TAddPopupItemType Enumeration, page 630)
Not documented.

TBlendMode(see TBlendMode Enumeration, page 631)
Not documented.

TVTGetCellIsEmptyEvent(see TVTGetCellIsEmptyEvent Type, page 631)
Not documented.

TChangeReason(see TChangeReason Enumeration, page 631)
Not documented.

TCheckImageKind(see TCheckImageKind Enumeration, page 632)
Determines which images should be used for checkboxes and radio buttons.

TCheckState(see TCheckState Enumeration, page 634)
Returns the current state of a node's check box, radio button or node button.

TCheckType(see TCheckType Enumeration, page 634)
Not documented.

TDragOperation(see TDragOperation Enumeration, page 635)
Not documented.

TVTGetImageExEvent(see TVTGetImageExEvent Type, page 635)
Not documented.

TDropMode(see TDropMode Enumeration, page 635)
Not documented.

THeaderState(see THeaderState Enumeration, page 636)
Not documented.

THintAnimationType(see THintAnimationType Enumeration, page 636)
Not documented.

THitPosition(see THitPosition Enumeration, page 637)
Not documented.

TItemEraseAction(see TItemEraseAction Enumeration, page 637)
Not documented.

TScrollBarStyle(see TScrollBarStyle Enumeration, page 638)
Not documented.

TSortDirection(see TSortDirection Enumeration, page 638)
Not documented.

TVirtualNodeInitState(see TVirtualNodeInitState Enumeration, page 638)
Not documented.

TVirtualNodeState(see TVirtualNodeState Enumeration, page 639)
Not documented.

TVirtualTreeColumnStyle(see TVirtualTreeColumnStyle Enumeration, page 640)
Not documented.

TVSTTextSourceType(see TVSTTextSourceType Enumeration, page 640)
Not documented.

TVSTTextType(see TVSTTextType Enumeration, page 640)
Not documented.

TVTAnimationOption(see TVTAnimationOption Enumeration, page 641)
Not documented.

TVTAutoOption(see TVTAutoOption Enumeration, page 641)
Not documented.

TVTButtonFillMode(see TVTButtonFillMode Enumeration, page 642)
Determines how the interior of nodes buttons should be drawn.

TVTButtonStyle(see TVTButtonStyle Enumeration, page 643)
Not documented.

TVTColumnOption(see TVTColumnOption Enumeration, page 643)
Not documented.

TVTDragImageKind(see TVTDragImageKind Enumeration, page 644)
Not documented.

TVTDragMoveRestriction(see TVTDragMoveRestriction Enumeration, page 644)
Not documented.

TVTDragType(see TVTDragType Enumeration, page 644)
Not documented.

TVTDrawSelectionMode(see TVTDrawSelectionMode Enumeration, page 645)
Not documented.

TVTDropMarkMode(see TVTDropMarkMode Enumeration, page 645)
Not documented.

TVTHeaderColumnLayout(see TVTHeaderColumnLayout Enumeration, page 645)

Not documented.

TVTHeaderOption(see TVTHeaderOption Enumeration, page 646)
Not documented.

TVTHeaderPopupOption(see TVTHeaderPopupOption Enumeration, page 646)
Not documented.

TVTMenuItem(see TVTMenuItem Type, page 647)
Not documented.

TVTHeaderStyle(see TVTHeaderStyle Enumeration, page 647)
Not documented.

TVTHintMode(see TVTHintMode Enumeration, page 648)
Not documented.

TVTImageInfoIndex(see TVTImageInfoIndex Enumeration, page 648)
Not documented.

TVTImageKind(see TVTImageKind Enumeration, page 648)
Not documented.

TVTIncrementalSearch(see TVTIncrementalSearch Enumeration, page 649)
Not documented.

TVTInternalPaintOption(see TVTInternalPaintOption Enumeration, page 649)
Not documented.

TVTLineMode(see TVTLineMode Enumeration, page 650)
Not documented.

TVTLineStyle(see TVTLineStyle Enumeration, page 650)
Not documented.

TVTLineType(see TVTLineType Enumeration, page 651)
Not documented.

TVTMiscOption(see TVTMiscOption Enumeration, page 651)
Not documented.

TVTNodeAlignment(see TVTNodeAlignment Enumeration, page 652)
Not documented.

TVTNodeAttachMode(see TVTNodeAttachMode Enumeration, page 652)
Not documented.

TVTScrollbarShowEvent(see TVTScrollbarShowEvent Type, page 653)
Not documented.

TVTPaintOption(see TVTPaintOption Enumeration, page 653)
Not documented.

TVTSearchDirection(see TVTSearchDirection Enumeration, page 654)
Not documented.

TVTSearchStart(see TVTSearchStart Enumeration, page 654)
Not documented.

TVTSelectionOption(see TVTSelectionOption Enumeration, page 655)
Not documented.

TVTStringOption(see TVTStringOption Enumeration, page 655)
Not documented.

TVTUpdateState(see TVTUpdateState Enumeration, page 656)
Not documented.

Variables

CF_CSV(see CF_CSV Variable, page 658)
Not documented.

CF_HTML(see CF_HTML Variable, page 658)
Not documented.

CF_VIRTUALTREE(see CF_VIRTUALTREE Variable, page 658)

Not documented.

CF_VRTF(see CF_VRTF Variable, page 659)
Not documented.

CF_VRTFNOOBJS(see CF_VRTFNOOBJS Variable, page 659)
Not documented.

CF_VTREFERENCE(see CF_VTREFERENCE Variable, page 659)
Not documented.

ClipboardDescriptions(see ClipboardDescriptions Variable, page 660)
Not documented.

DarkCheckImages(see DarkCheckImages Variable, page 660)
Not documented.

DarkTickImages(see DarkTickImages Variable, page 660)
Not documented.

FlatImages(see FlatImages Variable, page 661)
Not documented.

HintFont(see HintFont Variable, page 661)
Not documented.

HintWindowDestroyed(see HintWindowDestroyed Variable, page 661)
Not documented.

Initialized(see Initialized Variable, page 662)
Not documented.

InternalClipboardFormats(see InternalClipboardFormats Variable, page 662)
Not documented.

IsWin2K(see IsWin2K Variable, page 662)
Not documented.

IsWinNT(see IsWinNT Variable, page 663)
Not documented.

IsWinXP(see IsWinXP Variable, page 663)
Not documented.

LightCheckImages(see LightCheckImages Variable, page 663)
Not documented.

LightTickImages(see LightTickImages Variable, page 664)
Not documented.

MMXAvailable(see MMXAvailable Variable, page 664)
Not documented.

NeedToUnitialize(see NeedToUnitialize Variable, page 664)
Not documented.

StandardOLEFormat(see StandardOLEFormat Variable, page 665)
Not documented.

SystemCheckImages(see SystemCheckImages Variable, page 665)
Not documented.

SystemFlatCheckImages(see SystemFlatCheckImages Variable, page 665)
Not documented.

UtilityImages(see UtilityImages Variable, page 666)
Not documented.

Watcher(see Watcher Variable, page 666)
Not documented.

WorkerThread(see WorkerThread Variable, page 666)
Not documented.

WorkEvent(see WorkEvent Variable, page 667)
Not documented.

XPImages(see XPImages Variable, page 667)
Not documented.

Legend

Class

Function

Struct

Type

Variable

Constant

10.1 Classes

These are all classes that are contained in this documentation.

Classes

EVirtualTreeError(see EVirtualTreeError Class, page 87)

TBaseVirtualTree(see TBaseVirtualTree Class, page 88)
TBaseVirtualTree is the main and base class for all other Virtual Treeview descendants.

TBufferedString(see TBufferedString Class, page 241)

TClipboardFormatList(see TClipboardFormatList Class, page 243)
Not documented.

TClipboardFormats(see TClipboardFormats Class, page 246)
List of strings describing clipboard formats.

TCriticalSection(see TCriticalSection Class, page 248)
Not documented.

TCustomStringTreeOptions(see TCustomStringTreeOptions Class, page 250)
Enhanced options class for string trees.

TCustomVirtualDrawTree(see TCustomVirtualDrawTree Class, page 252)
Simple owner draw descendant of the base tree.

TCustomVirtualStringTree(see TCustomVirtualStringTree Class, page 274)
Descendant of TBaseVirtualTree(see TBaseVirtualTree Class, page 88), which is able to manage node captions on its own

TCustomVirtualTreeOptions(see TCustomVirtualTreeOptions Class, page 308)
Organizes all tree options into subproperties for easier managment.

TEnumFormatEtc(see TEnumFormatEtc Class, page 311)

TScrollBarOptions(see TScrollBarOptions Class, page 313)

TStringEditLink(see TStringEditLink Class, page 316)
TStringEditLink is the standard node editor of a TVirtualStringTree(see TVirtualStringTree Class, page 402).

TStringTreeOptions(see TStringTreeOptions Class, page 320)
Options class used in the string tree and its descentants.

TVirtualDrawTree(see TVirtualDrawTree Class, page 323)
Descendant of TBaseVirtualTree(see TBaseVirtualTree Class, page 88), which passes node paint events through to the application (similar to a draw grid)

TVirtualStringTree(see TVirtualStringTree Class, page 402)
Descentant of TBaseVirtualTree(see TBaseVirtualTree Class, page 88) which is able to manage node captions on its own.

TVirtualTreeColumn(see TVirtualTreeColumn Class, page 485)
Represents a column in a Virtual Treeview.

TVirtualTreeColumns(see TVirtualTreeColumns Class, page 497)
Collection class, which holds the columns for the tree.

TVirtualTreeHintWindow(see TVirtualTreeHintWindow Class, page 510)
Internally used hint window class to support Unicode hints.

TVirtualTreeOptions(see TVirtualTreeOptions Class, page 513)
Collects all binary options of the tree control into one place for easier access.

TVTColors(see TVTColors Class, page 515)
Collects all color related options for the tree control.

TVTDataObject(see TVTDataObject Class, page 521)
Implementation of an IDataObject interface.

TVTDragImage(see TVTDragImage Class, page 529)
Not documented.

TVTDragManager(see TVTDragManager Class, page 536)
Not documented.

TVTEdit(see TVTEdit Class, page 539)
Not documented.

TVTHeader(see TVTHeader Class, page 544)
Not documented.

TVTHeaderPopupMenu(see TVTHeaderPopupMenu Class, page 557)
Not documented.

TWideBufferedString(see TWideBufferedString Class, page 560)
Not documented.

TWorkerThread(see TWorkerThread Class, page 561)
Not documented.

TWriterHack(see TWriterHack Class, page 564)
Not documented.

Group

Virtual Treeview(see page 71)

Legend

Class

10.1.1 EVirtualTreeError Class

```
EVirtualTreeError = class(Exception);
```

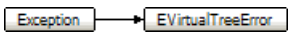
Description

EVirtualTreeError is a normal exception derivation especially for Virtual Treeview. This class does not add much value to its parent class but is rather there to better tell when an exception particularly from Virtual Treeview was raised.

Group

Classes(see page 86)

Class Hierarchy



File

VirtualTrees

10.1.2 TBaseVirtualTree Class

TBaseVirtualTree is the main and base class for all other Virtual Treeview descendants.

Pascal

```
TBaseVirtualTree = class(TCustomControl);
```

Description

This class implements most of the base features and abilities and can be used to derive new classes, which want to hide most of the details of the tree, which other descendants like TVirtualStringTree(see TVirtualStringTree Class, page 402) publish. Do not use the base treeview as object. It is not meant to be instantiated directly, instead via an descendant.

Events

OnAdvancedHeaderDraw(see TBaseVirtualTree.OnAdvancedHeaderDraw Event, page 128)
Header paint support event.

OnAfterCellPaint(see TBaseVirtualTree.OnAfterCellPaint Event, page 128)
Paint support event.

OnAfterItemErase(see TBaseVirtualTree.OnAfterItemErase Event, page 129)
Paint support event.

OnAfterItemPaint(see TBaseVirtualTree.OnAfterItemPaint Event, page 129)
Paint support event.

OnAfterPaint(see TBaseVirtualTree.OnAfterPaint Event, page 129)
Paint support event.

OnBeforeCellPaint(see TBaseVirtualTree.OnBeforeCellPaint Event, page 130)
Paint support event.

OnBeforeItemErase(see TBaseVirtualTree.OnBeforeItemErase Event, page 130)
Paint support event.

OnBeforeItemPaint(see TBaseVirtualTree.OnBeforeItemPaint Event, page 130)
Paint support event.

OnBeforePaint(see TBaseVirtualTree.OnBeforePaint Event, page 131)
Paint support event.

OnChange(see TBaseVirtualTree.OnChange Event, page 131)
Navigation support event.

OnChecked(see TBaseVirtualTree.OnChecked Event, page 131)
Check support event.

OnChecking(see TBaseVirtualTree.OnChecking Event, page 131)
Check support event.

OnCollapsed(see TBaseVirtualTree.OnCollapsed Event, page 132)
Miscellaneous event.

OnCollapsing(see TBaseVirtualTree.OnCollapsing Event, page 132)
Miscellaneous event.

OnColumnClick(see TBaseVirtualTree.OnColumnClick Event, page 132)
Header and column support event.

OnColumnDblClick(see TBaseVirtualTree.OnColumnDblClick Event, page 132)
Header and column support event.

OnColumnResize(see TBaseVirtualTree.OnColumnResize Event, page 133)
Header and column support routine.

OnCompareNodes(see TBaseVirtualTree.OnCompareNodes Event, page 133)
Sort and search support event.

OnCreateDataObject(see TBaseVirtualTree.OnCreateDataObject Event, page 134)
Drag'n drop support event.

OnCreateDragManager(see TBaseVirtualTree.OnCreateDragManager Event, page 134)
Drag'n drop support event.

OnCreateEditor(see TBaseVirtualTree.OnCreateEditor Event, page 134)
Editing support event.

OnDragAllowed(see TBaseVirtualTree.OnDragAllowed Event, page 135)
Drag'n drop support event.

OnDragDrop(see TBaseVirtualTree.OnDragDrop Event, page 135)
Drag'n drop support event.

OnDragOver(see TBaseVirtualTree.OnDragOver Event, page 137)
Drag'n drop support event.

OnEditCancelled(see TBaseVirtualTree.OnEditCancelled Event, page 137)
Editing support event.

OnEdited(see TBaseVirtualTree.OnEdited Event, page 137)
Editing support event.

OnEditing(see TBaseVirtualTree.OnEditing Event, page 138)
Editing support event.

OnExpanded(see TBaseVirtualTree.OnExpanded Event, page 138)
Misscellaneous event.

OnExpanding(see TBaseVirtualTree.OnExpanding Event, page 138)
Miscellaneous event.

OnFocusChanged(see TBaseVirtualTree.OnFocusChanged Event, page 138)
Navigation support event.

OnFocusChanging(see TBaseVirtualTree.OnFocusChanging Event, page 139)
Navigation support event.

OnFreeNode(see TBaseVirtualTree.OnFreeNode Event, page 139)
Data management node.

OnGetCellIsEmpty(see TBaseVirtualTree.OnGetCellIsEmpty Event, page 139)
Triggered when the tree control needs to know whether a given column is empty.

OnGetCursor(see TBaseVirtualTree.OnGetCursor Event, page 140)
Miscellaneous event.

OnGetHeaderCursor(see TBaseVirtualTree.OnGetHeaderCursor Event, page 140)
Header and column support event.

OnGetHelpContext(see TBaseVirtualTree.OnGetHelpContext Event, page 140)
Miscellaneous event.

OnGetImageIndex(see TBaseVirtualTree.OnGetImageIndex Event, page 140)
Display management event.

OnGetImageIndexEx(see TBaseVirtualTree.OnGetImageIndexEx Event, page 141)
Not documented.

OnGetLineStyle(see TBaseVirtualTree.OnGetLineStyle Event, page 141)
Display management event.

OnGetNodeDataSize(see TBaseVirtualTree.OnGetNodeDataSize Event, page 142)
Data management event.

OnGetPopupMenu(see TBaseVirtualTree.OnGetPopupMenu Event, page 142)
Miscellaneous event.

OnGetUserClipboardFormats(see TBaseVirtualTree.OnGetUserClipboardFormats Event, page 142)
Drag'n drop and clipboard support event.

OnHeaderClick(see TBaseVirtualTree.OnHeaderClick Event, page 143)
Header & column support event.

OnHeaderDblClick(see TBaseVirtualTree.OnHeaderDblClick Event, page 143)

Header & column support event.

OnHeaderDragged(see TBaseVirtualTree.OnHeaderDragged Event, page 143)
Header & column support event.

OnHeaderDraggedOut(see TBaseVirtualTree.OnHeaderDraggedOut Event, page 144)
Header & column support event.

OnHeaderDragging(see TBaseVirtualTree.OnHeaderDragging Event, page 144)
Header & column support event.

OnHeaderDraw(see TBaseVirtualTree.OnHeaderDraw Event, page 144)
Header & column support event.

OnHeaderDrawQueryElements(see TBaseVirtualTree.OnHeaderDrawQueryElements Event, page 145)
Header & column support event.

OnHeaderMouseDown(see TBaseVirtualTree.OnHeaderMouseDown Event, page 145)
Header & column support event.

OnHeaderMouseMove(see TBaseVirtualTree.OnHeaderMouseMove Event, page 145)
Header & column support event.

OnHeaderMouseUp(see TBaseVirtualTree.OnHeaderMouseUp Event, page 145)
Header & column support event.

OnHotChange(see TBaseVirtualTree.OnHotChange Event, page 146)
Navigation support event.

OnIncrementalSearch(see TBaseVirtualTree.OnIncrementalSearch Event, page 146)
Miscellaneous event.

OnInitChildren(see TBaseVirtualTree.OnInitChildren Event, page 147)
Node management event.

OnInitNode(see TBaseVirtualTree.OnInitNode Event, page 147)
Node management event.

OnKeyAction(see TBaseVirtualTree.OnKeyAction Event, page 148)
Miscellaneous event.

OnLoadNode(see TBaseVirtualTree.OnLoadNode Event, page 148)
Streaming support event.

OnMeasureItem(see TBaseVirtualTree.OnMeasureItem Event, page 148)
Miscellaneous event.

OnNodeCopied(see TBaseVirtualTree.OnNodeCopied Event, page 149)
Miscellaneous event.

OnNodeCopying(see TBaseVirtualTree.OnNodeCopying Event, page 149)
Miscellaneous event.

OnNodeMoved(see TBaseVirtualTree.OnNodeMoved Event, page 149)
Miscellaneous event.

OnNodeMoving(see TBaseVirtualTree.OnNodeMoving Event, page 150)
Miscellaneous event.

OnPaintBackground(see TBaseVirtualTree.OnPaintBackground Event, page 150)
Paint support event.

OnRenderOLEData(see TBaseVirtualTree.OnRenderOLEData Event, page 150)
Drag'n drop and clipboard support event.

OnResetNode(see TBaseVirtualTree.OnResetNode Event, page 151)
Node management event.

OnSaveNode(see TBaseVirtualTree.OnSaveNode Event, page 151)
Streaming support event.

OnScroll(see TBaseVirtualTree.OnScroll Event, page 152)
Miscellaneous event.

OnShowScrollbar(see TBaseVirtualTree.OnShowScrollbar Event, page 152)
Not documented.

OnStateChange(see TBaseVirtualTree.OnStateChange Event, page 152)
Miscellaneous event.

OnStructureChange(see TBaseVirtualTree.OnStructureChange Event, page 152)
Miscellaneous event.

OnUpdating(see TBaseVirtualTree.OnUpdating Event, page 153)
Miscellaneous event.

Group

Classes(see page 86)

Methods

AbsoluteIndex(see TBaseVirtualTree.AbsoluteIndex Method , page 160)
Reads the overall index of a node.

AddChild(see TBaseVirtualTree.AddChild Method , page 160)
Creates and adds a new child node to given node.

AddFromStream(see TBaseVirtualTree.AddFromStream Method , page 161)
Adds the content from the given stream to the given node.

AddToSelection(see TBaseVirtualTree.AddToSelection Method (PVirtualNode), page 161)
Adds one or more nodes to the current selection.

AdjustPaintCellRect(see TBaseVirtualTree.AdjustPaintCellRect Method , page 162)
Used in descentants to modify the clip rectangle of the current column while painting a certain node.

AdjustPanningCursor(see TBaseVirtualTree.AdjustPanningCursor Method , page 162)
Loads the proper cursor which indicates into which direction scrolling is done.

AdviseChangeEvent(see TBaseVirtualTree.AdviseChangeEvent Method , page 162)
Used to register a delayed change event.

AllocateInternalDataArea(see TBaseVirtualTree.AllocateInternalDataArea Method , page 163)
Registration method to allocate tree internal data per node.

Animate(see TBaseVirtualTree.Animate Method , page 163)
Support method for animated actions in the tree view.

Assign(see TBaseVirtualTree.Assign Method , page 164)
Used to copy properties from another Virtual Treeview.

BeginDrag(see TBaseVirtualTree.BeginDrag Method , page 164)
Starts an OLE drag'n drop operation.

BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164)
Enters the tree into a special synchronized mode.

BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164)
Locks the tree view to perform several update operations.

CalculateSelectionRect(see TBaseVirtualTree.CalculateSelectionRect Method , page 165)
Support method for draw selection.

CanAutoScroll(see TBaseVirtualTree.CanAutoScroll Method , page 165)
Determines whether the tree can currently auto scroll its window.

CancelCutOrCopy(see TBaseVirtualTree.CancelCutOrCopy Method , page 165)
Canceles any pending cut or copy clipboard operation.

CancelEditNode(see TBaseVirtualTree.CancelEditNode Method , page 166)
Cancel the current edit operation, if there is any.

CanEdit(see TBaseVirtualTree.CanEdit Method , page 166)
Determines whether a node can be edited or not.

CanFocus(see TBaseVirtualTree.CanFocus Method , page 166)
Support method to determine whether the tree window can receive the input focus.

CanShowDragImage(see TBaseVirtualTree.CanShowDragImage Method , page 166)
Determines whether a drag image should be shown.

Change(see TBaseVirtualTree.Change Method , page 167)

Central method called when a node's selection state changes.

ChangeScale(see TBaseVirtualTree.ChangeScale Method , page 167)
Helper method called by the VCL when control resizing is due.

CheckParentCheckState(see TBaseVirtualTree.CheckParentCheckState Method , page 167)
Helper method for recursive check state changes.

Clear(see TBaseVirtualTree.Clear Method , page 168)
Clears the tree and removes all nodes.

ClearChecked(see TBaseVirtualTree.ClearChecked Method , page 168)
Not documented.

ClearSelection(see TBaseVirtualTree.ClearSelection Method , page 168)
Removes all nodes from the current selection.

ClearTempCache(see TBaseVirtualTree.ClearTempCache Method , page 168)
Helper method to clear(see TBaseVirtualTree.Clear Method , page 168) the internal temporary node cache.

ColumnIsEmpty(see TBaseVirtualTree.ColumnIsEmpty Method , page 169)
Used to determine if a cell is considered as being empty.

CopyTo(see TBaseVirtualTree.CopyTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 169)
Copies Source and all its child nodes to Target.

CopyToClipBoard(see TBaseVirtualTree.CopyToClipBoard Method , page 169)
Copies all currently selected nodes to the clipboard.

CountLevelDifference(see TBaseVirtualTree.CountLevelDifference Method , page 170)
Determines the level difference of two nodes.

CountVisibleChildren(see TBaseVirtualTree.CountVisibleChildren Method , page 170)
Determines the number of visible child nodes of the given node.

Create(see TBaseVirtualTree.Create Constructor , page 170)
Constructor of the control

CreateParams(see TBaseVirtualTree.CreateParams Method , page 171)
Prepares the creation of the controls window handle.

CreateWnd(see TBaseVirtualTree.CreateWnd Method , page 171)
Initializes data, which depends on the window handle.

CutToClipBoard(see TBaseVirtualTree.CutToClipBoard Method , page 171)
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties(see TBaseVirtualTree.DefineProperties Method , page 171)
Helper method to customize loading and saving persistent tree data.

DeleteChildren(see TBaseVirtualTree.DeleteChildren Method , page 172)
Removes all child nodes from the given node.

DeleteNode(see TBaseVirtualTree.DeleteNode Method , page 172)
Removes the given node from the tree.

DeleteSelectedNodes(see TBaseVirtualTree.DeleteSelectedNodes Method , page 172)
Removes all currently selected nodes form the tree.

Destroy(see TBaseVirtualTree.Destroy Destructor , page 173)
Destructor of the control.

DetermineHiddenChildrenFlag(see TBaseVirtualTree.DetermineHiddenChildrenFlag Method , page 173)
Determines whether all children of a given node are hidden.

DetermineHiddenChildrenFlagAllNodes(see TBaseVirtualTree.DetermineHiddenChildrenFlagAllNodes Method , page 173)
Determines whether all children of all nodes are hidden.

DetermineHitPositionLTR(see TBaseVirtualTree.DetermineHitPositionLTR Method , page 174)
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineHitPositionRTL(see TBaseVirtualTree.DetermineHitPositionLTR Method , page 174)
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineNextCheckState(see TBaseVirtualTree.DetermineNextCheckState Method , page 174)
Not documented.

DetermineScrollDirections(see TBaseVirtualTree.DetermineScrollDirections Method , page 174)
Not documented.

DoAdvancedHeaderDraw(see TBaseVirtualTree.DoAdvancedHeaderDraw Method , page 174)
Not documented.

DoAfterCellPaint(see TBaseVirtualTree.DoAfterCellPaint Method , page 175)
Not documented.

DoAfterItemErase(see TBaseVirtualTree.DoAfterItemErase Method , page 175)
Not documented.

DoAfterItemPaint(see TBaseVirtualTree.DoAfterItemPaint Method , page 175)
Not documented.

DoAfterPaint(see TBaseVirtualTree.DoAfterPaint Method , page 175)
Not documented.

DoAutoScroll(see TBaseVirtualTree.DoAutoScroll Method , page 176)
Enables or disables the auto scroll timer.

DoBeforeCellPaint(see TBaseVirtualTree.DoBeforeCellPaint Method , page 176)
Not documented.

DoBeforeDrag(see TBaseVirtualTree.DoBeforeDrag Method , page 176)
Not documented.

DoBeforeItemErase(see TBaseVirtualTree.DoBeforeItemErase Method , page 177)
Not documented.

DoBeforeItemPaint(see TBaseVirtualTree.DoBeforeItemPaint Method , page 177)
Not documented.

DoBeforePaint(see TBaseVirtualTree.DoBeforePaint Method , page 177)
Not documented.

DoCancelEdit(see TBaseVirtualTree.DoCancelEdit Method , page 177)
Called when the tree should stop editing without accepting changed values.

DoCanEdit(see TBaseVirtualTree.DoCanEdit Method , page 178)
Not documented.

DoChange(see TBaseVirtualTree.DoChange Method , page 178)
Not documented.

DoCheckClick(see TBaseVirtualTree.DoCheckClick Method , page 178)
Not documented.

DoChecked(see TBaseVirtualTree.DoChecked Method , page 178)
Not documented.

DoChecking(see TBaseVirtualTree.DoChecking Method , page 179)
Not documented.

DoCollapsed(see TBaseVirtualTree.DoCollapsed Method , page 179)
Not documented.

DoCollapsing(see TBaseVirtualTree.DoCollapsing Method , page 179)
Not documented.

DoColumnClick(see TBaseVirtualTree.DoColumnClick Method , page 179)
Not documented.

DoColumnDblClick(see TBaseVirtualTree.DoColumnDblClick Method , page 180)
Not documented.

DoColumnResize(see TBaseVirtualTree.DoColumnResize Method , page 180)
Not documented.

DoCompare(see TBaseVirtualTree.DoCompare Method , page 180)
Not documented.

DoCreateDataObject(see TBaseVirtualTree.DoCreateDataObject Method , page 180)
Not documented.

DoCreateDragManager(see TBaseVirtualTree.DoCreateDragManager Method , page 181)

Not documented.

DoCreateEditor(see TBaseVirtualTree.DoCreateEditor Method , page 181)
Not documented.

DoDragDrop(see TBaseVirtualTree.DoDragDrop Method , page 181)
Not documented.

DoDragExpand(see TBaseVirtualTree.DoDragExpand Method , page 181)
Not documented.

DoDragging(see TBaseVirtualTree.DoDragging Method , page 182)
Internal method which handles drag' drop.

DoDragOver(see TBaseVirtualTree.DoDragOver Method , page 182)
Not documented.

DoEdit(see TBaseVirtualTree.DoEdit Method , page 182)
Initiates editing of the currently set focused column and edit node.

DoEndDrag(see TBaseVirtualTree.DoEndDrag Method , page 182)
Not documented.

DoEndEdit(see TBaseVirtualTree.DoEndEdit Method , page 183)
Stops the current edit operation and takes over the new content.

DoExpanded(see TBaseVirtualTree.DoExpanded Method , page 183)
Not documented.

DoExpanding(see TBaseVirtualTree.DoExpanding Method , page 183)
Not documented.

DoFocusChange(see TBaseVirtualTree.DoFocusChange Method , page 184)
Not documented.

DoFocusChanging(see TBaseVirtualTree.DoFocusChanging Method , page 184)
Not documented.

DoFocusNode(see TBaseVirtualTree.DoFocusNode Method , page 184)
Internal method to set the focused node.

DoFreeNode(see TBaseVirtualTree.DoFreeNode Method , page 184)
Not documented.

DoGetAnimationType(see TBaseVirtualTree.DoGetAnimationType Method , page 185)
Determines the type of animation to be used.

DoGetCursor(see TBaseVirtualTree.DoGetCursor Method , page 185)
Not documented.

DoGetHeaderCursor(see TBaseVirtualTree.DoGetHeaderCursor Method , page 185)
Not documented.

DoGetImageIndex(see TBaseVirtualTree.DoGetImageIndex Method , page 186)
Not documented.

DoGetLineStyle(see TBaseVirtualTree.DoGetLineStyle Method , page 186)
Not documented.

DoGetNodeHint(see TBaseVirtualTree.DoGetNodeHint Method , page 186)
Not documented.

DoGetNodeTooltip(see TBaseVirtualTree.DoGetNodeTooltip Method , page 186)
Not documented.

DoGetNodeWidth(see TBaseVirtualTree.DoGetNodeWidth Method , page 187)
Overridable method which always retuns 0.

DoGetPopupMenu(see TBaseVirtualTree.DoGetPopupMenu Method , page 187)
Overridable method which triggers the OnGetPopup event.

DoGetUserClipboardFormats(see TBaseVirtualTree.DoGetUserClipboardFormats Method , page 187)
Not documented.

DoHeaderClick(see TBaseVirtualTree.DoHeaderClick Method , page 187)
Not documented.

DoHeaderDblClick(see TBaseVirtualTree.DoHeaderDblClick Method , page 188)
Not documented.

DoHeaderDragged(see TBaseVirtualTree.DoHeaderDragged Method , page 188)
Not documented.

DoHeaderDraggedOut(see TBaseVirtualTree.DoHeaderDraggedOut Method , page 188)
Not documented.

DoHeaderDragging(see TBaseVirtualTree.DoHeaderDragging Method , page 189)
Not documented.

DoHeaderDraw(see TBaseVirtualTree.DoHeaderDraw Method , page 189)
Not documented.

DoHeaderDrawQueryElements(see TBaseVirtualTree.DoHeaderDrawQueryElements Method , page 189)
Not documented.

DoHeaderMouseDown(see TBaseVirtualTree.DoHeaderMouseDown Method , page 189)
Not documented.

DoHeaderMouseMove(see TBaseVirtualTree.DoHeaderMouseMove Method , page 190)
Not documented.

DoHeaderMouseUp(see TBaseVirtualTree.DoHeaderMouseUp Method , page 190)
Not documented.

DoHotChange(see TBaseVirtualTree.DoHotChange Method , page 190)
Not documented.

DoIncrementalSearch(see TBaseVirtualTree.DoIncrementalSearch Method , page 190)
Not documented.

DoInitChildren(see TBaseVirtualTree.DoInitChildren Method , page 191)
Not documented.

DoInitNode(see TBaseVirtualTree.DoInitNode Method , page 191)
Not documented.

DoKeyAction(see TBaseVirtualTree.DoKeyAction Method , page 191)
Not documented.

DoLoadUserData(see TBaseVirtualTree.DoLoadUserData Method , page 191)
Not documented.

DoMeasureItem(see TBaseVirtualTree.DoMeasureItem Method , page 192)
Not documented.

DoNodeCopied(see TBaseVirtualTree.DoNodeCopied Method , page 192)
Not documented.

DoNodeCopying(see TBaseVirtualTree.DoNodeCopying Method , page 192)
Not documented.

DoNodeMoved(see TBaseVirtualTree.DoNodeMoved Method , page 192)
Not documented.

DoNodeMoving(see TBaseVirtualTree.DoNodeMoving Method , page 193)
Not documented.

DoPaintBackground(see TBaseVirtualTree.DoPaintBackground Method , page 193)
Not documented.

DoPaintDropMark(see TBaseVirtualTree.DoPaintDropMark Method , page 193)
Overridable method which draws the small line on top of a nodes image depending on the current drop state.

DoPaintNode(see TBaseVirtualTree.DoPaintNode Method , page 193)
Overridable method which does nothing.

DoPopupMenu(see TBaseVirtualTree.DoPopupMenu Method , page 194)
Overridable method which shows the popup menu for the given node.

DoRenderOLEData(see TBaseVirtualTree.DoRenderOLEData Method , page 194)
Not documented.

DoReset(see TBaseVirtualTree.DoReset Method , page 194)

Not documented.

DoSaveUserData(see TBaseVirtualTree.DoSaveUserData Method , page 194)
Not documented.

DoScroll(see TBaseVirtualTree.DoScroll Method , page 195)
Overridable method which triggers the OnScroll(see TBaseVirtualTree.OnScroll Event, page 152) event.

DoSetOffsetXY(see TBaseVirtualTree.DoSetOffsetXY Method , page 195)
Internal core routine to set the tree's scroll position.

DoShowScrollbar(see TBaseVirtualTree.DoShowScrollbar Method , page 195)
Not documented.

DoStartDrag(see TBaseVirtualTree.DoStartDrag Method , page 196)
Not documented.

DoStateChange(see TBaseVirtualTree.DoStateChange Method , page 196)
Not documented.

DoStructureChange(see TBaseVirtualTree.DoStructureChange Method , page 196)
Not documented.

DoTimerScroll(see TBaseVirtualTree.DoTimerScroll Method , page 196)
Callback method which is triggered whenever the scroll timer fires.

DoUpdating(see TBaseVirtualTree.DoUpdating Method , page 197)
Not documented.

DoValidateCache(see TBaseVirtualTree.DoValidateCache Method , page 197)
Not documented.

DragCanceled(see TBaseVirtualTree.DragCanceled Method , page 197)
Called by the VCL when a drag'n drop operation was canceled by the user.

DragDrop(see TBaseVirtualTree.DragDrop Method , page 197)
Helper method, which is used when a drag operation is finished.

DragEnter(see TBaseVirtualTree.DragEnter Method , page 198)
Not documented.

DragFinished(see TBaseVirtualTree.DragFinished Method , page 198)
Called when a drag operation is finished (accepted or cancelled).

Dragging(see TBaseVirtualTree.Dragging Method , page 198)
Returns true if a drag'n drop operation is in progress.

DragLeave(see TBaseVirtualTree.DragLeave Method , page 199)
Not documented.

DragOver(see TBaseVirtualTree.DragOver Method , page 199)
Not documented.

DrawDottedHLine(see TBaseVirtualTree.DrawDottedHLine Method , page 199)
Not documented.

DrawDottedVLine(see TBaseVirtualTree.DrawDottedVLine Method , page 199)
Not documented.

EditNode(see TBaseVirtualTree.EditNode Method , page 200)
Starts editing the given node if allowed to.

EndEditNode(see TBaseVirtualTree.EndEditNode Method , page 200)
Stops node editing if it was started before.

EndSynch(see TBaseVirtualTree.EndSynch Method , page 200)
Counterpart to BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164).

EndUpdate(see TBaseVirtualTree.EndUpdate Method , page 201)
Resets the update lock set by BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164).

ExecuteAction(see TBaseVirtualTree.ExecuteAction Method , page 201)
Not documented.

FindNodeInSelection(see TBaseVirtualTree.FindNodeInSelection Method , page 201)
Helper method to find the given node in the current selection.

FinishChunkHeader(see TBaseVirtualTree.FinishChunkHeader Method , page 201)
Not documented.

FinishCutOrCopy(see TBaseVirtualTree.FinishCutOrCopy Method , page 202)
Stops any pending cut or copy clipboard operation.

FlushClipboard(see TBaseVirtualTree.FlushClipboard Method , page 202)
Renders all pending clipboard data.

FontChanged(see TBaseVirtualTree.FontChanged Method , page 202)
Not documented.

FullCollapse(see TBaseVirtualTree.FullCollapse Method , page 203)
Collapses all nodes in the tree.

FullExpand(see TBaseVirtualTree.FullExpand Method , page 203)
Expands all nodes in the tree.

GetBorderDimensions(see TBaseVirtualTree.GetBorderDimensions Method , page 203)
Not documented.

GetCheckImage(see TBaseVirtualTree.GetCheckImage Method , page 203)
Not documented.

GetCheckImageListFor(see TBaseVirtualTree.GetCheckImageListFor Method , page 204)
Not documented.

GetColumnClass(see TBaseVirtualTree.GetColumnClass Method , page 204)
Returns the class to be used to manage columns in the tree.

GetControlsAlignment(see TBaseVirtualTree.GetControlsAlignment Method , page 204)
Not documented.

GetDisplayRect(see TBaseVirtualTree.GetDisplayRect Method , page 205)
Returns the visible region used by the given node in client coordinates.

GetFirst(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstChecked(see TBaseVirtualTree.GetFirstChecked Method , page 206)
Not documented.

GetFirstChild(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstCutCopy(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstInitialized(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstSelected(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisible(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleChild(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleChildNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetHeaderClass(see TBaseVirtualTree.GetHeaderClass Method , page 206)
Returns the header class to be used by the tree.

GetHintWindowClass(see TBaseVirtualTree.GetHintWindowClass Method , page 206)
Not documented.

GetHitTestInfoAt(see TBaseVirtualTree.GetHitTestInfoAt Method , page 206)

Returns information about the node at the given position.

GetImageIndex(see TBaseVirtualTree.GetImageIndex Method , page 207)
Not documented.

GetLast(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastChild(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastChildNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastInitialized(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisible(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleChild(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleChildNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetMaxColumnWidth(see TBaseVirtualTree.GetMaxColumnWidth Method , page 208)
Returns the width of the largest node in the given column.

GetMaxRightExtend(see TBaseVirtualTree.GetMaxRightExtend Method , page 208)
Determines the maximum with of the currently visible part of the tree.

GetNativeClipboardFormats(see TBaseVirtualTree.GetNativeClipboardFormats Method , page 208)
Used to let descendants and the application add their own supported clipboard formats.

GetNext(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextChecked(see TBaseVirtualTree.GetNextChecked Method , page 209)
Not documented.

GetNextCutCopy(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextInitialized(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextSelected(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextSibling(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisible(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleSibling(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleSiblingNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNodeAt(see TBaseVirtualTree.GetNodeAt Method (Integer, Integer), page 210)
Not documented.

GetNodeData(see TBaseVirtualTree.GetNodeData Method , page 210)
Returns the address of the user data area of the given node.

GetNodeLevel(see TBaseVirtualTree.GetNodeLevel Method , page 210)
Returns the indentation level of the given node.

GetOptionsClass(see TBaseVirtualTree.GetOptionsClass Method , page 211)
Customization helper to determine which options class the tree should use.

GetPrevious(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousInitialized(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousSibling(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisible(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleSibling(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleSiblingNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetSortedCutCopySet(see TBaseVirtualTree.GetSortedCutCopySet Method , page 212)
Returns a sorted list of nodes, which are marked for s cut or copy clipboard operation.

GetSortedSelection(see TBaseVirtualTree.GetSortedSelection Method , page 213)
Returns a sorted list of all currently selected nodes.

GetTextInfo(see TBaseVirtualTree.GetTextInfo Method , page 213)
Helper method for node editors, hints etc.

GetTreeFromDataObject(see TBaseVirtualTree.GetTreeFromDataObject Method , page 213)
OLE drag'n drop and clipboard support method.

GetTreeRect(see TBaseVirtualTree.GetTreeRect Method , page 214)
Returns the size of the virtual tree image.

GetVisibleParent(see TBaseVirtualTree.GetVisibleParent Method , page 214)
Returns the first (nearest) parent node, which is visible.

HandleHotTrack(see TBaseVirtualTree.HandleHotTrack Method , page 214)
Not documented.

HandleIncrementalSearch(see TBaseVirtualTree.HandleIncrementalSearch Method , page 214)
Not documented.

HandleMouseDblClick(see TBaseVirtualTree.HandleMouseDblClick Method , page 215)
Not documented.

HandleMouseDown(see TBaseVirtualTree.HandleMouseDown Method , page 215)
Not documented.

HandleMouseUp(see TBaseVirtualTree.HandleMouseUp Method , page 215)
Not documented.

HasAsParent(see TBaseVirtualTree.HasAsParent Method , page 215)
Determines if the given node has got another node as one of its parents.

HasImage(see TBaseVirtualTree.HasImage Method , page 216)
Not documented.

HasPopupMenu(see TBaseVirtualTree.HasPopupMenu Method , page 216)
Determines whether there is a pop up menu assigned to the tree.

InitChildren(see TBaseVirtualTree.InitChildren Method , page 216)

Not documented.

InitNode(see TBaseVirtualTree.InitNode Method , page 217)
Not documented.

InsertNode(see TBaseVirtualTree.InsertNode Method , page 217)
Inserts a new node and returns it to the caller.

InternalAddFromStream(see TBaseVirtualTree.InternalAddFromStream Method , page 217)
Not documented.

InternalAddToSelection(see TBaseVirtualTree.InternalAddToSelection Method (PVirtualNode, Boolean), page 218)
Not documented.

InternalCacheNode(see TBaseVirtualTree.InternalCacheNode Method , page 218)
Not documented.

InternalClearSelection(see TBaseVirtualTree.InternalClearSelection Method , page 218)
Not documented.

InternalConnectNode(see TBaseVirtualTree.InternalConnectNode Method , page 219)
Not documented.

InternalData(see TBaseVirtualTree.InternalData Method , page 219)
Returns the address of the internal data for a tree class.

InternalDisconnectNode(see TBaseVirtualTree.InternalDisconnectNode Method , page 219)
Not documented.

InternalRemoveFromSelection(see TBaseVirtualTree.InternalRemoveFromSelection Method , page 220)
Not documented.

InvalidateCache(see TBaseVirtualTree.InvalidateCache Method , page 220)
Empties the internal node cache and marks it as invalid.

InvalidateChildren(see TBaseVirtualTree.InvalidateChildren Method , page 220)
Invalidates all children of the given node.

InvalidateColumn(see TBaseVirtualTree.InvalidateColumn Method , page 220)
Invalidates the client area part of a column.

InvalidateNode(see TBaseVirtualTree.InvalidateNode Method , page 221)
Invalidates the given node.

InvalidateToBottom(see TBaseVirtualTree.InvalidateToBottom Method , page 221)
Invalidates the client area starting with the top position of the given node.

InvertSelection(see TBaseVirtualTree.InvertSelection Method , page 221)
Inverts the current selection.

IsEditing(see TBaseVirtualTree.IsEditing Method , page 222)
Tells the caller whether the tree is currently in edit mode.

IsMouseSelecting(see TBaseVirtualTree.IsMouseSelecting Method , page 222)
Tell the caller whether the tree is currently in draw selection mode.

IterateSubtree(see TBaseVirtualTree.IterateSubtree Method , page 222)
Iterator method to go through all nodes of a given sub tree.

Loaded(see TBaseVirtualTree.Loaded Method , page 223)
Not documented.

LoadFromFile(see TBaseVirtualTree.LoadFromFile Method , page 223)
Loads previously streamed out tree data back in again.

LoadFromStream(see TBaseVirtualTree.LoadFromFile Method , page 223)
Loads previously streamed out tree data back in again.

MainColumnChanged(see TBaseVirtualTree.MainColumnChanged Method , page 223)
Not documented.

MarkCutCopyNodes(see TBaseVirtualTree.MarkCutCopyNodes Method , page 223)
Not documented.

MeasureItemHeight(see TBaseVirtualTree.MeasureItemHeight Method , page 224)
Not documented.

MouseMove(see TBaseVirtualTree.MouseMove Method , page 224)
Not documented.

MoveTo(see TBaseVirtualTree.MoveTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 224)
Moves Source and all its child nodes to Target.

Notification(see TBaseVirtualTree.Notification Method , page 225)
Not documented.

OriginalWMNCPaint(see TBaseVirtualTree.OriginalWMNCPaint Method , page 225)
Not documented.

Paint(see TBaseVirtualTree.Paint Method , page 225)
TControl's Paint method used here to display the tree.

PaintCheckImage(see TBaseVirtualTree.PaintCheckImage Method , page 225)
Not documented.

PaintImage(see TBaseVirtualTree.PaintImage Method , page 226)
Not documented.

PaintNodeButton(see TBaseVirtualTree.PaintNodeButton Method , page 226)
Not documented.

PaintSelectionRectangle(see TBaseVirtualTree.PaintSelectionRectangle Method , page 226)
Not documented.

PaintTree(see TBaseVirtualTree.PaintTree Method , page 226)
Main paint routine for the tree image.

PaintTreeLines(see TBaseVirtualTree.PaintTreeLines Method , page 227)
Not documented.

PanningWindowProc(see TBaseVirtualTree.PanningWindowProc Method , page 227)
Not documented.

PasteFromClipboard(see TBaseVirtualTree.PasteFromClipboard Method , page 227)
Inserts the content of the clipboard into the tree.

PrepareDragImage(see TBaseVirtualTree.PrepareDragImage Method , page 228)
Not documented.

Print(see TBaseVirtualTree.Print Method , page 228)
Not documented.

ProcessDrop(see TBaseVirtualTree.ProcessDrop Method , page 228)
Helper method to ease OLE drag'n drop operations.

ProcessOLEData(see TBaseVirtualTree.ProcessOLEData Method , page 229)
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk(see TBaseVirtualTree.ReadChunk Method , page 229)
Not documented.

ReadNode(see TBaseVirtualTree.ReadNode Method , page 229)
Not documented.

RedirectFontChangeEvent(see TBaseVirtualTree.RedirectFontChangeEvent Method , page 230)
Not documented.

ReinitChildren(see TBaseVirtualTree.ReinitChildren Method , page 230)
Forces all child nodes of Node to be reinitialized.

ReinitNode(see TBaseVirtualTree.ReinitNode Method , page 230)
Forces a reinitialization of the given node.

RemoveFromSelection(see TBaseVirtualTree.RemoveFromSelection Method , page 230)
Removes the given node from the current selection.

RenderOLEData(see TBaseVirtualTree.RenderOLEData Method , page 231)
Renders pending OLE data.

RepaintNode(see TBaseVirtualTree.RepaintNode Method , page 231)
Causes the treeview to repaint the given node.

ResetNode(see TBaseVirtualTree.ResetNode Method , page 231)

Resets the given node to uninitialized.

ResetRangeAnchor(see TBaseVirtualTree.ResetRangeAnchor Method , page 232)
Not documented.

RestoreFontChangeEvent(see TBaseVirtualTree.RestoreFontChangeEvent Method , page 232)
Not documented.

SaveToFile(see TBaseVirtualTree.SaveToFile Method , page 232)
Saves the entire content of the tree into a file or stream.

SaveToStream(see TBaseVirtualTree.SaveToFile Method , page 232)
Saves the entire content of the tree into a file or stream.

ScrollIntoView(see TBaseVirtualTree.ScrollIntoView Method , page 232)
Scrolls the tree so that the given node comes in the client area.

SelectAll(see TBaseVirtualTree.SelectAll Method , page 233)
Selects all nodes in the tree.

SelectNodes(see TBaseVirtualTree.SelectNodes Method , page 233)
Selects a range of nodes.

SetBiDiMode(see TBaseVirtualTree.SetBiDiMode Method , page 233)
Not documented.

SetFocusedNodeAndColumn(see TBaseVirtualTree.SetFocusedNodeAndColumn Method , page 234)
Not documented.

SkipNode(see TBaseVirtualTree.SkipNode Method , page 234)
Not documented.

Sort(see TBaseVirtualTree.Sort Method , page 234)
Sorts the given node.

SortTree(see TBaseVirtualTree.SortTree Method , page 234)
Sorts the entire tree view.

StartWheelPanning(see TBaseVirtualTree.StartWheelPanning Method , page 235)
Not documented.

StopWheelPanning(see TBaseVirtualTree.StopWheelPanning Method , page 235)
Not documented.

StructureChange(see TBaseVirtualTree.StructureChange Method , page 235)
Not documented.

SuggestDropEffect(see TBaseVirtualTree.SuggestDropEffect Method , page 236)
Not documented.

ToggleNode(see TBaseVirtualTree.ToggleNode Method , page 236)
Changes a node's expand state to the opposite state.

ToggleSelection(see TBaseVirtualTree.ToggleSelection Method , page 236)
Toggles the selection state of a range of nodes.

UnselectNodes(see TBaseVirtualTree.UnselectNodes Method , page 236)
Deselects a range of nodes.

UpdateAction(see TBaseVirtualTree.UpdateAction Method , page 237)
Not documented.

UpdateDesigner(see TBaseVirtualTree.UpdateDesigner Method , page 237)
Not documented.

UpdateEditBounds(see TBaseVirtualTree.UpdateEditBounds Method , page 237)
Not documented.

UpdateHeaderRect(see TBaseVirtualTree.UpdateHeaderRect Method , page 237)
Not documented.

UpdateHorizontalScrollBar(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateScrollBars(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateVerticalScrollBar(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateWindowAndDragImage(see TBaseVirtualTree.UpdateWindowAndDragImage Method , page 238)
Not documented.

UseRightToLeftReading(see TBaseVirtualTree.UseRightToLeftReading Method , page 238)
Helper method for right-to-left layout.

ValidateCache(see TBaseVirtualTree.ValidateCache Method , page 239)
Initiates the validation of the internal node cache.

ValidateChildren(see TBaseVirtualTree.ValidateChildren Method , page 239)
Validates all children of a given node.

ValidateNode(see TBaseVirtualTree.ValidateNode Method , page 239)
Validates a given node.

ValidateNodeDataSize(see TBaseVirtualTree.ValidateNodeDataSize Method , page 239)
Helper method for node data size initalization.

WndProc(see TBaseVirtualTree.WndProc Method , page 240)
Redirected window procedure to do some special processing.

WriteChunks(see TBaseVirtualTree.WriteChunks Method , page 240)
Writes the core chunks for the given node to the given stream.

WriteNode(see TBaseVirtualTree.WriteNode Method , page 241)
Writes the cover (envelop) chunk for the given node to the given stream.

Properties

Alignment(see TBaseVirtualTree.Alignment Property, page 107)
Determines the horizontal alignment of text if no columns are defined.

AnimationDuration(see TBaseVirtualTree.AnimationDuration Property, page 107)
Determines the maximum duration the tree can use to play an animation.

AutoExpandDelay(see TBaseVirtualTree.AutoExpandDelay Property, page 107)
Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay(see TBaseVirtualTree.AutoScrollDelay Property, page 108)
Time which determines when auto scrolling should start.

AutoScrollInterval(see TBaseVirtualTree.AutoScrollInterval Property, page 108)
Time interval between scroll events when doing auto scroll.

Background(see TBaseVirtualTree.Background Property, page 108)
Holds a background image for the tree.

BackgroundOffsetX(see TBaseVirtualTree.BackgroundOffsetX Property, page 109)
Horizontal offset of the background image.

BackgroundOffsetY(see TBaseVirtualTree.BackgroundOffsetY Property, page 109)
Vertical offset of the background image.

BorderStyle(see TBaseVirtualTree.BorderStyle Property, page 109)
Same as TForm.BorderStyle.

ButtonFillMode(see TBaseVirtualTree.ButtonFillMode Property, page 109)
Determines how to fill the background of the node buttons.

ButtonStyle(see TBaseVirtualTree.ButtonStyle Property, page 110)
Determines the look of node buttons.

ChangeDelay(see TBaseVirtualTree.ChangeDelay Property, page 110)
Time which determines when the OnChange(see TBaseVirtualTree.OnChange Event, page 131) event should be triggered after the actual change event.

CheckImageKind(see TBaseVirtualTree.CheckImageKind Property, page 110)
Determines which images should be used for checkboxes and radio buttons.

CheckImages(see TBaseVirtualTree.CheckImages Property, page 111)
Not documented.

CheckState(see TBaseVirtualTree.CheckState Property, page 111)
Read or set the check state of a node.

CheckType(see TBaseVirtualTree.CheckType Property, page 111)
Read or set the check type of a node.

ChildCount(see TBaseVirtualTree.ChildCount Property, page 111)
Read or set the number of child nodes of a node.

ChildrenInitialized(see TBaseVirtualTree.ChildrenInitialized Property, page 112)
Read whether a node's child count has been initialized already.

ClipboardFormats(see TBaseVirtualTree.ClipboardFormats Property, page 112)
Special class to keep a list of clipboard format descriptions.

Colors(see TBaseVirtualTree.Colors Property, page 112)
A collection of colors used in the tree.

CustomCheckImages(see TBaseVirtualTree.CustomCheckImages Property, page 113)
Assign your own image list to get the check images you like most.

DefaultNodeHeight(see TBaseVirtualTree.DefaultNodeHeight Property, page 113)
Read or set the height new nodes get as initial value.

DefaultPasteMode(see TBaseVirtualTree.DefaultPasteMode Property, page 113)
Read or set the value, which determines where to add pasted nodes to.

DragHeight(see TBaseVirtualTree.DragHeight Property, page 114)
Read or set the vertical limit of the internal drag image.

DragImage(see TBaseVirtualTree.DragImage Property, page 114)
Holds the instance of the internal drag image.

DragImageKind(see TBaseVirtualTree.DragImageKind Property, page 114)
Read or set what should be shown in the drag image.

DragManager(see TBaseVirtualTree.DragManager Property, page 114)
Holds the reference to the internal drag manager.

DragOperations(see TBaseVirtualTree.DragOperations Property, page 115)
Read or set which drag operations may be allowed in the tree.

DragSelection(see TBaseVirtualTree.DragSelection Property, page 115)
Keeps a temporary list of nodes during drag'n drop.

DragType(see TBaseVirtualTree.DragType Property, page 115)
Read or set which subsystem should be used for dragging(see TBaseVirtualTree.Dragging Method , page 198).

DragWidth(see TBaseVirtualTree.DragWidth Property, page 116)
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode(see TBaseVirtualTree.DrawSelectionMode Property, page 116)
Read or set how multiselection with the mouse is to be visualized.

DropTargetNode(see TBaseVirtualTree.DropTargetNode Property, page 116)
Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

EditColumn(see TBaseVirtualTree.EditColumn Property, page 117)
Not documented.

EditDelay(see TBaseVirtualTree.EditDelay Property, page 117)
Read or set the maximum time between two single clicks on the same node, which should start node editing.

EditLink(see TBaseVirtualTree.EditLink Property, page 117)
Keeps a reference to the internal edit link during a node edit operation.

Expanded(see TBaseVirtualTree.Expanded Property, page 118)
Read or set the expanded state of a particular node.

FocusedColumn(see TBaseVirtualTree.FocusedColumn Property, page 118)
Read or set the currently focused collumn.

FocusedNode(see TBaseVirtualTree.FocusedNode Property, page 118)
Read or set the currently focused node.

Font(see TBaseVirtualTree.Font Property, page 119)
Same as TWinControl.Font.

FullyVisible(see TBaseVirtualTree.FullyVisible Property, page 119)

Read or set whether a node is fully visible or not.

HasChildren(see TBaseVirtualTree.HasChildren Property, page 119)
Read or set whether a node has got children.

Header(see TBaseVirtualTree.Header Property, page 120)
Provides access to the header instance.

HeaderRect(see TBaseVirtualTree.HeaderRect Property, page 120)
Returns the non-client-area rectangle used for the header.

HintAnimation(see TBaseVirtualTree.HintAnimation Property, page 120)
Read or set the current hint animation type.

HintMode(see TBaseVirtualTree.HintMode Property, page 121)
Read or set what type of hint you want for the tree view.

HotCursor(see TBaseVirtualTree.HotCursor Property, page 121)
Read or set which cursor should be used for hot nodes.

HotNode(see TBaseVirtualTree.HotNode Property, page 121)
Read, which node is currently the hot node.

Images(see TBaseVirtualTree.Images Property, page 122)
Read or set the tree's normal image list.

IncrementalSearch(see TBaseVirtualTree.IncrementalSearch Property, page 122)
Read or set the current incremental search mode.

IncrementalSearchDirection(see TBaseVirtualTree.IncrementalSearchDirection Property, page 122)
Read or set the direction to be used for incremental search.

IncrementalSearchStart(see TBaseVirtualTree.IncrementalSearchStart Property, page 123)
Read or set where to start incremental search.

IncrementalSearchTimeout(see TBaseVirtualTree.IncrementalSearchTimeout Property, page 123)
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Indent(see TBaseVirtualTree.Indent Property, page 123)
Read or set the indentation amount for node levels.

IsDisabled(see TBaseVirtualTree.IsDisabled Property, page 124)
Read or set the enabled state of the given node.

IsVisible(see TBaseVirtualTree.IsVisible Property, page 124)
Read or set the visibility state of the given node.

LastClickPos(see TBaseVirtualTree.LastClickPos Property, page 124)
Used for retained drag start and wheel mouse scrolling.

LastDropMode(see TBaseVirtualTree.LastDropMode Property, page 125)
Read how the last drop operation finished.

LineMode(see TBaseVirtualTree.LineMode Property, page 125)
Read or set the mode of the tree lines.

LineStyle(see TBaseVirtualTree.LineStyle Property, page 125)
Read or set the mode of the tree lines.

Margin(see TBaseVirtualTree.Margin Property, page 125)
Read or set the tree's node margin.

MultiLine(see TBaseVirtualTree.MultiLine Property, page 126)
Read or toggle the multiline feature for a given node.

NodeAlignment(see TBaseVirtualTree.NodeAlignment Property, page 126)
Read or set the node alignment value.

NodeDataSize(see TBaseVirtualTree.NodeDataSize Property, page 127)
Read or set the extra data size for each node.

NodeHeight(see TBaseVirtualTree.NodeHeight Property, page 127)
Read or set a node's height.

NodeParent(see TBaseVirtualTree.NodeParent Property, page 127)
Read or set a node's parent node.

OffsetX(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

OffsetXY(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

OffsetY(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

RootNode(see TBaseVirtualTree.RootNode Property, page 153)
Reference to the internal root node which is the anchor of the entire tree node hierarchy.

RootNodeCount(see TBaseVirtualTree.RootNodeCount Property, page 153)
Read or set the number of nodes on the top level.

ScrollBarOptions(see TBaseVirtualTree.ScrollBarOptions Property, page 154)
Reference to the scroll bar options class.

SearchBuffer(see TBaseVirtualTree.SearchBuffer Property, page 154)
Current input string for incremental search.

Selected(see TBaseVirtualTree.Selected Property, page 154)
Property to modify or determine the selection state of a node.

SelectedCount(see TBaseVirtualTree.SelectedCount Property, page 155)
Contains the number of selected nodes.

SelectionBlendFactor(see TBaseVirtualTree.SelectionBlendFactor Property, page 155)
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

SelectionCurveRadius(see TBaseVirtualTree.SelectionCurveRadius Property, page 155)
Read or set the current corner radius for node selection rectangles.

StateImages(see TBaseVirtualTree.StateImages Property, page 156)
Reference to the images list which is used for the state images.

TextMargin(see TBaseVirtualTree.TextMargin Property, page 156)
Read or set the distance of the node caption to its borders.

TopNode(see TBaseVirtualTree.TopNode Property, page 157)
The top node is the node which is currently at the top border of the client area.

TotalCount(see TBaseVirtualTree.TotalCount Property, page 157)
Returns the number of nodes in the tree.

TotalInternalDataSize(see TBaseVirtualTree.TotalInternalDataSize Property, page 157)
Keeps the currently accumulated data size for one node.

TreeOptions(see TBaseVirtualTree.TreeOptions Property, page 158)
Reference to the tree's options.

TreeStates(see TBaseVirtualTree.TreeStates Property, page 158)
Property which keeps a set of flags which indicate current operation and states of the tree.

UpdateCount(see TBaseVirtualTree.UpdateCount Property, page 158)
Not documented.

VerticalAlignment(see TBaseVirtualTree.VerticalAlignment Property, page 158)
Used to set a node's vertical button aligment with regard to the entire node rectangle.

VisibleCount(see TBaseVirtualTree.VisibleCount Property, page 159)
Number of currently visible nodes.

VisiblePath(see TBaseVirtualTree.VisiblePath Property, page 159)
Property to set or determine a node parent's expand states.

WantTabs(see TBaseVirtualTree.WantTabs Property, page 159)
Read or set whether the tree wants to process tabs on its own.

Legend

protected

Property


public

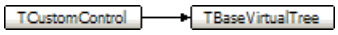

read only


Event


Method


virtual

Class Hierarchy



File

VirtualTrees

10.1.2.1 TBaseVirtualTree.Alignment Property

Determines the horizontal alignment of text if no columns are defined.

Pascal

```
property Alignment: TAlignment;
```

Description

This property is only used if there are no columns defined and applies only to the node captions. Right alignment means here the right client area border and left aligned means the node buttons/lines etc. (both less the text margin).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.2 TBaseVirtualTree.AnimationDuration Property

Determines the maximum duration the tree can use to play an animation.

Pascal

```
property AnimationDuration: Cardinal;
```

Description

The value is specified in milliseconds and per default there are 200 ms as time frame, which is the recommended duration for such operations. On older systems (particularly Windows 95 and Windows 98) the animation process might not get enough CPU time to avoid expensive animations to finish properly. Still the animation loop tries to stay as close as possible to the given time.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.3 TBaseVirtualTree.AutoExpandDelay Property

Time delay after which a node gets expanded if it is the current drop target.

Pascal

```
property AutoExpandDelay: Cardinal;
```

Description

This value is specified in milliseconds and determines when to expand a node if it is the current drop target. This value is only used if voAutoDropExpand in Options is set.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.4 TBaseVirtualTree.AutoScrollDelay Property

Time which determines when auto scrolling should start.

Pascal

```
property AutoScrollDelay: Cardinal;
```

Description

Once the mouse pointer has been moved near to a border a timer is started using the interval specified by AutoScrollDelay. When the timer has fired auto scrolling starts provided it is enabled (see also TreeOptions(🏠 see TBaseVirtualTree.TreeOptions Property, page 158)). The value is specified in milliseconds.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.5 TBaseVirtualTree.AutoScrollInterval Property

Time interval between scroll events when doing auto scroll.

Pascal

```
property AutoScrollInterval: TAutoScrollInterval;
```

Description

This property determines the speed how the tree is scrolled vertically or horizontally when auto scrolling is in progress. The value is given in milliseconds.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.6 TBaseVirtualTree.Background Property

Holds a background image for the tree.

Pascal

```
property Background: TPicture;
```

Description

Virtual Treeview supports a fixed background image which does not scroll but can be adjusted by BackgroundOffsetX(🏠 see TBaseVirtualTree.BackgroundOffsetX Property, page 109) and BackgroundOffsetY(🏠 see TBaseVirtualTree.BackgroundOffsetY Property, page 109).

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.7 TBaseVirtualTree.BackgroundOffsetX Property

Horizontal offset of the background image.

Pascal

```
property BackgroundOffsetX: Integer;
```

Description

Determines the horizontal offset of the left border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.8 TBaseVirtualTree.BackgroundOffsetY Property

Vertical offset of the background image.

Pascal

```
property BackgroundOffsetY: Integer;
```

Description

Determines the vertical offset of the top border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.9 TBaseVirtualTree.BorderStyle Property

Same as TForm.BorderStyle.

Pascal

```
property BorderStyle: TBorderStyle;
```

Description

See TForm.BorderStyle.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.10 TBaseVirtualTree.ButtonFillMode Property

Determines how to fill the background of the node buttons.

Pascal

```
property ButtonFillMode: TVTButtonFillMode;
```

Description

This property is used to specify how the interior of the little plus and minus node buttons should be drawn, if ButtonStyle(see TBaseVirtualTree.ButtonStyle Property, page 110) is bsTriangle.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.11 TBaseVirtualTree.ButtonStyle Property

Determines the look of node buttons.

Pascal

```
property ButtonStyle: TVTButtonStyle;
```

Description

Determines the look of node buttons.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.12 TBaseVirtualTree.ChangeDelay Property

Time which determines when the OnChange(see TBaseVirtualTree.OnChange Event, page 131) event should be triggered after the actual change event.

Pascal

```
property ChangeDelay: Cardinal;
```

Description

In order to accumulate many quick changes in the tree you can use this delay value to specify after which wait time the OnChange(see TBaseVirtualTree.OnChange Event, page 131) event should occur. A value of 0 means to trigger OnChange(see TBaseVirtualTree.OnChange Event, page 131) immediately after the change (usually a selection or focus change) happend. Any value > 0 will start a timer which then triggers OnChange(see TBaseVirtualTree.OnChange Event, page 131).

Note that there is the synchronous mode (started by BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164)) which effectively circumvents the change delay for the duration of the synchronous mode (stopped by EndSynch(see TBaseVirtualTree.EndSynch Method , page 200)) regardless of the ChangeDelay setting.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.13 TBaseVirtualTree.CheckImageKind Property

Determines which images should be used for checkboxes and radio buttons.

Pascal

```
property CheckImageKind: TCheckImageKind;
```

Description

CheckImageKind can be used to switch the image set, which should be used for the tree. Read the description about TCheckImageKind(see TCheckImageKind Enumeration, page 632) for a list of all images, which can be used. CheckImageKind can also be set to ckCustom, which allows to supply a customized set of images to the tree. In order to have that working you must assign an image list (TCustomImageList) to the CustomCheckImages(see

TBaseVirtualTree.CustomCheckImages Property, page 113) property.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.14 TBaseVirtualTree.CheckImages Property

Not documented.

Pascal

```
property CheckImages: TCustomImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.15 TBaseVirtualTree.CheckState Property

Read or set the check state of a node.

Pascal

```
property CheckState [Node: PVirtualNode]: TCheckState;
```

Description

The CheckState property can be used to read the current check state of a node or to set a new one. Virtual Treeview ensures that invalid check states (e.g. csMixedPressed for radio buttons) do not cause an error.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.16 TBaseVirtualTree.CheckType Property

Read or set the check type of a node.

Pascal

```
property CheckType [Node: PVirtualNode]: TCheckType;
```

Description

The CheckType property can be used to read the current check type of a node or to set a new one. Setting a new check type will reset a the node's check state to csUncheckedNormal.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.17 TBaseVirtualTree.ChildCount Property

Read or set the number of child nodes of a node.

Pascal

```
property ChildCount [Node: PVirtualNode]: Cardinal;
```

Description

ChildCount can be used to read the current number of child nodes or to change it. Assigning a lower value than there was before will automatically delete as many child nodes (starting from the last child) as there are more than what was set. Increasing the value will add new child nodes. Note: code behind this property is very effective, so it using ChildCount is highly recommended over manipulating the child count using AddChild(see TBaseVirtualTree.AddChild Method , page 160), InsertNode(see TBaseVirtualTree.InsertNode Method , page 217) and DeleteNode(see TBaseVirtualTree.DeleteNode Method , page 172).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.18 TBaseVirtualTree.ChildrenInitialized Property

Read whether a node's child count has been initialized already.

Pascal

```
property ChildrenInitialized [Node: PVirtualNode]: Boolean;
```

Description

This read only property is used to determine whether a node's child count has been set. Alternatively, the child count value is not considered if vsHasChildren is not in the node states.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.19 TBaseVirtualTree.ClipboardFormats Property

Special class to keep a list of clipboard format descriptions.

Pascal

```
property ClipboardFormats: TClipboardFormats;
```

Description

This TStringList descendant is used to keep a number of clipboard format descriptions, which are usually used to register clipboard formats with the system. Using a string list for this task allows to store enabled clipboard formats in the DFM.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.20 TBaseVirtualTree.Colors Property

A collection of colors used in the tree.

Pascal

```
property Colors: TVTColors;
```

Description

This property holds an instance of the TVTColors(see TVTColors Class, page 515) class, which is used to customize many of the colors used in a tree. Placing them all in a specialized class helps organizing the colors in the object inspector and improves general management.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.21 TBaseVirtualTree.CustomCheckImages Property

Assign your own image list to get the check images you like most.

Pascal

```
property CustomCheckImages: TCustomImageList;
```

Description

The CustomCheckImages property is used when custom check images are enabled (see also ckCustom in TCheckImageKind(see TCheckImageKind Enumeration, page 632)).

See Also

TCheckImageKind(see TCheckImageKind Enumeration, page 632)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.22 TBaseVirtualTree.DefaultNodeHeight Property

Read or set the height new nodes get as initial value.

Pascal

```
property DefaultNodeHeight: Cardinal;
```

Description

This property allows to read the current initial height for new nodes and to set a new value. Note that changing the property value does not change the height of existing nodes. Only new nodes are affected.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.23 TBaseVirtualTree.DefaultPasteMode Property

Read or set the value, which determines where to add pasted nodes to.

Pascal

```
property DefaultPasteMode: TVTNodeAttachMode;
```

Description

The default paste mode is an attach mode, which is used when pasting data from the clipboard into the tree. Usually, you will want new nodes to be added as child nodes to the currently focused node (and this is also the default value), but you can also specify to add nodes only as siblings.

See Also

TVTNodeAttachMode(see TVTNodeAttachMode Enumeration, page 652)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.24 TBaseVirtualTree.DragHeight Property

Read or set the vertical limit of the internal drag image.

Pascal

```
property DragHeight: Integer;
```

Description

The DragHeight property (as well as the DragWidth(see TBaseVirtualTree.DragWidth Property, page 116) property) are only for compatibility reason in the tree. If a platform does not support the IDropTargetHelper(see IDropTargetHelper Interface, page 702) interface (Windows 9x/Me, Windows NT 4.0) then Virtual Treeview uses its own implementation of a DragImage(see TBaseVirtualTree.DragImage Property, page 114). Since displaying a translucent drag image is performance hungry you should limit the image size shown for the drag operation.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.25 TBaseVirtualTree.DragImage Property

Holds the instance of the internal drag image.

Pascal

```
property DragImage: TVTDragImage;
```

Description

For older systems where the IDropTargetHelper(see IDropTargetHelper Interface, page 702) interface is not supported Virtual Treeview simulates the translucent drag image during drag'n drop. The property DragImage makes the internal drag image instance accessible for special handling. The class itself is always created but is usually not visible when the IDropTargetHelper(see IDropTargetHelper Interface, page 702) interface is supported.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.26 TBaseVirtualTree.DragImageKind Property

Read or set what should be shown in the drag image.

Pascal

```
property DragImageKind: TVTDragImageKind;
```

Description

DragImageKind allows to switch parts of the drag image off and on.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.27 TBaseVirtualTree.DragManager Property

Holds the reference to the internal drag manager.

Pascal

```
property DragManager: IVTDragManager;
```

Description

The drag manager is the central point for the drag'n drop support in Virtual Treeview. Usually you do not need to access it but sometimes it might be necessary so the reference is accessible through this property.

See Also

TVTDragManager(see TVTDragManager Class, page 536)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.28 TBaseVirtualTree.DragOperations Property

Read or set which drag operations may be allowed in the tree.

Pascal

```
property DragOperations: TDragOperations;
```

Description

Using this property you can determine, which actions may be performed when a drag operation is finished. The default value includes move, copy and link, where link is rather an esoteric value and only there because it is supported by OLE. The values used directly determine which image is shown for the drag cursor. The specified drag operations do not tell which actions will actually be performed but only, which actions are allowed. They still can be modified during drag'n drop by using a modifier key like the control, shift or alt key or can entirely be ignored by the drop handler.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.29 TBaseVirtualTree.DragSelection Property

Keeps a temporary list of nodes during drag'n drop.

Pascal

```
property DragSelection: TNodeArray;
```

Description

This list is a local copy of the current selection array and is only used during a drag operation.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.30 TBaseVirtualTree.DragType Property

Read or set which subsystem should be used for dragging(see TBaseVirtualTree.Dragging Method , page 198).

Pascal

```
property DragType: TVTDragType;
```

Description

Traditionally, Delphi only supports its own drag mechanism, which is not compatible with the rest of the system. This VCL

dragging(🏠 see TBaseVirtualTree.Dragging Method , page 198) also does not support to transport random data nor does it support drag operations between applications. Thus Virtual Treeview also supports the generally used OLE dragging(🏠 see TBaseVirtualTree.Dragging Method , page 198), which in turn is incompatible with VCL dragging(🏠 see TBaseVirtualTree.Dragging Method , page 198). Depending on your needs you can enable either VCL or OLE dragging(🏠 see TBaseVirtualTree.Dragging Method , page 198) as both together cannot be started. However, Virtual Treeview is able to act as drop target for both kind of data, independant of what is set in DragType.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.31 TBaseVirtualTree.DragWidth Property

Read or set the horizontal limit of the internal drag image.

Pascal

```
property DragWidth: Integer;
```

Description

The DragWidth property (as well as the DragHeight(🏠 see TBaseVirtualTree.DragHeight Property, page 114) property) are only for compatibility reason in the tree. If a platform does not support the IDropTargetHelper(🏠 see IDropTargetHelper Interface, page 702) interface (Windows 9x/Me, Windows NT 4.0) then Virtual Treeview uses its own implementation of a DragImage(🏠 see TBaseVirtualTree.DragImage Property, page 114). Since displaying a translucent drag image is performance hungry you should limit the image size shown for the drag operation.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.32 TBaseVirtualTree.DrawSelectionMode Property

Read or set how multiselection with the mouse is to be visualized.

Pascal

```
property DrawSelectionMode: TVTDrawSelectionMode;
```

Description

Virtuall Treeview allows to display two different selection rectangles when doing multiselection with the mouse. One is the traditiional dotted focus rectangle and the other one is a translucent color rectangle. The latter is the preferred one but the former is set as default (for compatibility reasons).

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.33 TBaseVirtualTree.DropTargetNode Property

Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

Pascal

```
property DropTargetNode: PVirtualNode;
```

Description

The drop target node has no meaning except during drag'n drop and only if the tree it belongs to is itself the current drop target. But even then DropTargetNode might be nil, particularly when the mouse hovers over an area in the tree, which is

not covered by a node.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.34 TBaseVirtualTree.EditColumn Property

Not documented.

Pascal

```
property EditColumn: TColumnIndex;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.35 TBaseVirtualTree.EditDelay Property

Read or set the maximum time between two single clicks on the same node, which should start node editing.

Pascal

```
property EditDelay: Cardinal;
```

Description

A node edit operation can be started using the keyboard (F2 key), in code using EditNode(see TBaseVirtualTree.EditNode Method , page 200) or by clicking twice on the same node (but not doing a double click). EditDelay is the maxmimum time distance between both clicks in which the edit operation is started.

See Also

Editors and editing(see page 41)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.36 TBaseVirtualTree.EditLink Property

Keeps a reference to the internal edit link during a node edit operation.

Pascal

```
property EditLink: IVTEditLink;
```

Description

During an edit operation a link is established between the tree and the editor for the current node. By default a simple TEdit control is used as editor but due to the great customization possibilities there can be any node editor you may want. In order to communicate with this potentially unknown node editor the edit link is used. The EditLink property holds this link during the edit operation, so you can manipulate the interface.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.37 TBaseVirtualTree.Expanded Property

Read or set the expanded state of a particular node.

Pascal

```
property Expanded [Node: PVirtualNode]: Boolean;
```

Description

Using this property you can expand or collapse the given node. This method uses the central ToggleNode(see TBaseVirtualTree.ToggleNode Method , page 236) method.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.38 TBaseVirtualTree.FocusedColumn Property

Read or set the currently focused collumn.

Pascal

```
property FocusedColumn: TColumnIndex;
```

Description

When toExtendedFocus in TVTSelectionOptions(see TVTSelectionOptions Type, page 628) is enabled then the user can select node cells in others than the main column (the column with the tree structure). In order to keep track, which column is currently selected FocusedColumn is used (similar to FocusedNode(see TBaseVirtualTree.FocusedNode Property, page 118)).

See Also

FocusedNode(see TBaseVirtualTree.FocusedNode Property, page 118), TVTSelectionOptions(see TVTSelectionOptions Type, page 628)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.39 TBaseVirtualTree.FocusedNode Property

Read or set the currently focused node.

Pascal

```
property FocusedNode: PVirtualNode;
```

Description

One node (and only one) in the tree view can have the current input focus, marked as dotted rectangle around the node's caption. Having the input focus means this node can be edited by pressing F2 or clicking on it and user keyboard input is interpreted with respect to the focused node (e.g. tree navigation, expansion/collapsing etc.). If extended focus is enabled then also the FocusedColumn(see TBaseVirtualTree.FocusedColumn Property, page 118) property is taken into account. Read there for more info about column focus.

See Also

FocusedColumn(see TBaseVirtualTree.FocusedColumn Property, page 118), TVTSelectionOptions(see TVTSelectionOptions Type, page 628)

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.40 TBaseVirtualTree.Font Property

Same as TWinControl.Font.

Pascal

```
property Font;
```

Description

See TWinControl.Font.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.41 TBaseVirtualTree.FullyVisible Property

Read or set whether a node is fully visible or not.

Pascal

```
property FullyVisible [Node: PVirtualNode]: Boolean;
```

Description

Beside the fact that a node can be out of the client area there are two possibilities for it to be hidden. One is the vsVisible state in TVirtualNodeState(🏠 see TVirtualNodeState Enumeration, page 639), which hides the node regardles of the current state of another node, if not specified. The other one is that one or more parent nodes might be collapsed, hiding so their entire child nodes structure. The visibility flag itself can be checked using the IsVisible(🏠 see TBaseVirtualTree.IsVisible Property, page 124) property, while the expansion state of parents nodes can be examined via the VisiblePath(🏠 see TBaseVirtualTree.VisiblePath Property, page 159) property. If both are true then the node is said to be fully visible.

See Also

IsVisible(🏠 see TBaseVirtualTree.IsVisible Property, page 124), VisiblePath(🏠 see TBaseVirtualTree.VisiblePath Property, page 159), vsVisible, TVirtualNodeStates(🏠 see TVirtualNodeStates Type, page 598)

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.42 TBaseVirtualTree.HasChildren Property

Read or set whether a node has got children.

Pascal

```
property HasChildren [Node: PVirtualNode]: Boolean;
```

Description

A node can be set to have children by assigning true to this property. Internally this will add the vsHasChildren state to the node but not add any child nodes. This state in turn will cause the node to be drawn with a plus sign in front of its caption, denoting so it can be expanded and will show child nodes. As long as the child nodes are not touch in any way (e.g. by expanding the parent node or by navigatin or searching/sorting the tree) there will be no actual child nodes.

They simply do not exist yet. However they will be created as soon as an access is done.

Setting the HasChildren property to false will delete any existing child node.

See Also

vsHasChildren, TVirtualNodeStates(🏠 see TVirtualNodeStates Type, page 598)

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.43 TBaseVirtualTree.Header Property

Provides access to the header instance.

Pascal

```
property Header: TVTHeader;
```

Description

This property is used to allow access to the header instance, which manages all aspects of the tree's header image as well as the column settings.

See Also

TVTHeader(🏠 see TVTHeader Class, page 544)

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.44 TBaseVirtualTree.HeaderRect Property

Returns the non-client-area rectangle used for the header.

Pascal

```
property HeaderRect: TRect;
```

Description

Use this property to determine the extents used by the header of Virtual Treeview.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.45 TBaseVirtualTree.HintAnimation Property

Read or set the current hint animation type.

Pascal

```
property HintAnimation: THintAnimationType;
```

Description

With this property you can specify what animation you would like to play when displaying a hint. For some applications it might not be good to animate(🏠 see TBaseVirtualTree.Animate Method , page 163) hints, hence you can entirely switch them off. Usually however you will leave the system standard. This way the user can decide whether and which hint animation he or she likes.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.46 TBaseVirtualTree.HintMode Property

Read or set what type of hint you want for the tree view.

Pascal

```
property HintMode: TVTHintMode;
```

Description

Virtual Treeview supports several hints modes. This includes the normal hint used for any other TControl class as well as a node specific hint, which is individual for each node or even each cell.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.47 TBaseVirtualTree.HotCursor Property

Read or set which cursor should be used for hot nodes.

Pascal

```
property HotCursor: TCursor;
```

Description

When you enable toHotTrack in TreeOptions.PaintOptions then the node, which is currently under the mouse pointer becomes the hot node. This is a special state, which can be used for certain effects. Hot nodes have by default an underlined caption and may cause the cursor to change to what ever you like. The HotCursor property is used to specify, which cursor is to be used.

See Also

HotNode(see TBaseVirtualTree.HotNode Property, page 121), TVTPaintOptions(see TVTPaintOptions Type, page 625)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.48 TBaseVirtualTree.HotNode Property

Read, which node is currently the hot node.

Pascal

```
property HotNode: PVirtualNode;
```

Description

When you enable toHotTrack in TreeOptions.PaintOptions then the node, which is currently under the mouse pointer becomes the hot node. The property HotNode can be used to access this node for special handling.

See Also

HotCursor(see TBaseVirtualTree.HotCursor Property, page 121), toHotTrack, TVTPaintOptions(see TVTPaintOptions Type, page 625)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.49 TBaseVirtualTree.Images Property

Read or set the tree's normal image list.

Pascal

```
property Images: TCustomImageList;
```

Description

Just like with TListView and TTreeview also Virtual Treeview can take an image list for its normal images. Additionally, there are image lists for state images and check images.

See Also

StateImages(see TBaseVirtualTree.StateImages Property, page 156), CheckImages(see TBaseVirtualTree.CheckImages Property, page 111)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.50 TBaseVirtualTree.IncrementalSearch Property

Read or set the current incremental search mode.

Pascal

```
property IncrementalSearch: TVTIncrementalSearch;
```

Description

Virtual Treeview can do an incremental search by calling back the application when comparing node captions. The IncrementalSearch property determines whether incremental search is enabled and which nodes should be searched through.

See Also

IncrementalSearchDirection(see TBaseVirtualTree.IncrementalSearchDirection Property, page 122), IncrementalSearchStart(see TBaseVirtualTree.IncrementalSearchStart Property, page 123), IncrementalSearchTimeout(see TBaseVirtualTree.IncrementalSearchTimeout Property, page 123)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.51 TBaseVirtualTree.IncrementalSearchDirection Property

Read or set the direction to be used for incremental search.

Pascal

```
property IncrementalSearchDirection: TVTSearchDirection;
```

Description

When incremental search is enabled then Virtual Treeview can search forward and backward from the start point given by IncrementalSearchStart(see TBaseVirtualTree.IncrementalSearchStart Property, page 123).

See Also

IncrementalSearch(see TBaseVirtualTree.IncrementalSearch Property, page 122), IncrementalSearchStart(see TBaseVirtualTree.IncrementalSearchStart Property, page 123), IncrementalSearchTime123out

Class

TBaseVirtualTree Class(see page 88)

10.1.2.52 TBaseVirtualTree.IncrementalSearchStart Property

Read or set where to start incremental search.

Pascal

```
property IncrementalSearchStart: TVTSearchStart;
```

Description

When incremental search is enabled in the tree view then you can specify here, where to start the next incremental search operation from.

See Also

IncrementalSearch(see TBaseVirtualTree.IncrementalSearch Property, page 122), IncrementalSearchDirection(see TBaseVirtualTree.IncrementalSearchDirection Property, page 122), IncrementalSearchTimeout(see TBaseVirtualTree.IncrementalSearchTimeout Property, page 123)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.53 TBaseVirtualTree.IncrementalSearchTimeout Property

Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Pascal

```
property IncrementalSearchTimeout: Cardinal;
```

Description

When incremental search is enabled in Virtual Treeview then you can specify here after what time incremental search should stop when no keyboard input is encountered any longer. This property so determines also the speed at which users have to type letters to keep the incremental search rolling.

See Also

IncrementalSearch(see TBaseVirtualTree.IncrementalSearch Property, page 122), IncrementalSearchDirection(see TBaseVirtualTree.IncrementalSearchDirection Property, page 122), IncrementalSearchStart(see TBaseVirtualTree.IncrementalSearchStart Property, page 123)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.54 TBaseVirtualTree.Indent Property

Read or set the indentation amount for node levels.

Pascal

```
property Indent: Cardinal;
```

Description

Each new level in the tree (child nodes of a parent node) are visually shifted to distinguish betwenn them and their parent node (that's the tree layout after all). The Indent property determines the shift distance in pixels.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.55 TBaseVirtualTree.IsDisabled Property

Read or set the enabled state of the given node.

Pascal

```
property IsDisabled [Node: PVirtualNode]: Boolean;
```

Description

A node can have many different states. One of them is its enabled state, which can be set via this property. Enabling a node means it can be focused and selected, so it can take part in clipboard and drag'n drop operations, and can be edited.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.56 TBaseVirtualTree.IsVisible Property

Read or set the visibility state of the given node.

Pascal

```
property IsVisible [Node: PVirtualNode]: Boolean;
```

Description

A node can be made invisible using this property. That means, even if its parent nodes all are expanded the node is not shown and the visual image is as would the node not exist. However it still can be searched or take part in certain other operations.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.57 TBaseVirtualTree.LastClickPos Property

Used for retained drag start and wheel mouse scrolling.

Pascal

```
property LastClickPos: TPoint;
```

Description

This internal positions is made public to allow descendants to modify mainly the right click behavior of the tree control.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.58 TBaseVirtualTree.LastDropMode Property

Read how the last drop operation finished.

Pascal

```
property LastDropMode: TDropMode;
```

Description

In the case you don't handle drag'n drop operations directly in OnDragDrop(see TBaseVirtualTree.OnDragDrop Event, page 135) it might be necessary to know how the last drag operation finshed. Read more in the drag mode enumeration about what is possible.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.59 TBaseVirtualTree.LineMode Property

Read or set the mode of the tree lines.

Pascal

```
property LineMode: TVTLineMode;
```

Description

Apart from the usual lines Virtual Treeview also supports a special draw mode named bands. This allows for neat visual effects.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.60 TBaseVirtualTree.LineStyle Property

Read or set the mode of the tree lines.

Pascal

```
property LineStyle: TVTLineStyle;
```

Description

Virtual Treeview allows to customize the lines used to display the node hierarchy. The default style is a dotted pattern, but you can also make solid lines or specify your own line pattern.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.61 TBaseVirtualTree.Margin Property

Read or set the tree's node margin.

Pascal

```
property Margin: Integer;
```

Description

The node margin is the distance between the cell bounds and its content like the lines, images, check box and so on. However this border is only applied to the left and right side of the node cell.

Note: there is also a TextMargin(see TBaseVirtualTree.TextMargin Property, page 156) property in TVirtualStringTree(see TVirtualStringTree Class, page 402), which is an additional border for the cell text only.

See Also

TVirtualStringTree.TextMargin(see TVirtualStringTree.TextMargin Property, page 482)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.62 TBaseVirtualTree.MultiLine Property

Read or toggle the multiline feature for a given node.

Pascal

```
property MultiLine [Node: PVirtualNode]: Boolean;
```

Description

Since multiline support for nodes requires extra processing this behavior is switchable. When switched on the node is wrapped into the available space until the node height is exhausted. By including carriage return/line feed pairs you can explicitely specify where to start new lines. The node's height is not automatically adjusted to the given text. Instead there is an event (OnMeasureItem(see TBaseVirtualTree.OnMeasureItem Event, page 148)), which can be used to compute a node's height before it is displayed the first time. In addition an application can use the ComputeNodeHeight method to compute the height of the node depending on its caption text.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.63 TBaseVirtualTree.NodeAlignment Property

Read or set the node alignment value.

Pascal

```
property NodeAlignment: TVTNodeAlignment;
```

Description

Nodes have got an align member, which is used to determine the vertical position of the node's images and tree lines. The NodeAlignment property specifies how to interpret the value in the align member.

See Also

TVirtualNode(see TVirtualNode Record, page 576)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.64 TBaseVirtualTree.NodeDataSize Property

Read or set the extra data size for each node.

Pascal

```
property NodeDataSize: Integer;
```

Description

A node can have an area for user data, which can be used to store application defined, node specific data in. Use GetNodeData(see TBaseVirtualTree.GetNodeData Method , page 210) to get the address of this area. In addition to assigning a value here you can also use the OnGetNodeDataSize(see TBaseVirtualTree.OnGetNodeDataSize Event, page 142) event, which is called when NodeDataSize is -1.

See Also

Data handling(see page 39)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.65 TBaseVirtualTree.NodeHeight Property

Read or set a node's height.

Pascal

```
property NodeHeight [Node: PVirtualNode]: Cardinal;
```

Description

Each node can have its individual height, which is stored in the node's record. You could directly assign a value to this member but I strongly discourage this as it does not update certain other structures in the tree. Instead use the NodeHeight property here to modify a node's height.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.66 TBaseVirtualTree.NodeParent Property

Read or set a node's parent node.

Pascal

```
property NodeParent [Node: PVirtualNode]: PVirtualNode;
```

Description

When reading this property then either the node's real parent node is returned or nil if the parent node is the internal, hidden root node. When writing to this property you will effectively move a node to a new location.

See Also

MoveTo(see TBaseVirtualTree.MoveTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 224), CopyTo(see TBaseVirtualTree.CopyTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 169)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.67 TBaseVirtualTree.OffsetXY Property

Read or set the tree's current horizontal and vertical scroll offsets.

Pascal

```
property OffsetX: Integer;
property OffsetXY: TPoint;
property OffsetY: Integer;
```

Description

Virtual Treeview allows to retrieve or set the internal scroll offset directly, without sending WM_HSCROLL/WM_VSCROLL message around. This allows also to link two or more trees together. This scroll offset is given in pixels and is always less or equal 0.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.68 TBaseVirtualTree.OnAdvancedHeaderDraw Event

Header paint support event.

Pascal

```
property OnAdvancedHeaderDraw: TVTAdvancedHeaderPaintEvent;
```

Description

The OnAdvancedHeaderDraw event is used when owner draw is enabled for the header and a column is set to owner draw mode. It can be used to custom draw only certain parts of the header instead the whole thing. A good example for this event is customizing the background of the header for only one column. With the standard custom draw method (OnHeaderDraw(see TBaseVirtualTree.OnHeaderDraw Event, page 144)) you are in an all-or-nothing situation and have to paint everything in the header including the text, images and sort direction indicator. OnAdvancedHeaderDraw however uses OnHeaderDrawQueryElements(see TBaseVirtualTree.OnHeaderDrawQueryElements Event, page 145) to ask for the elements the application wants to draw and acts accordingly.

See Also

OnHeaderDrawQueryElements(see TBaseVirtualTree.OnHeaderDrawQueryElements Event, page 145), OnHeaderDraw(see TBaseVirtualTree.OnHeaderDraw Event, page 144)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.69 TBaseVirtualTree.OnAfterCellPaint Event

Paint support event.

Pascal

```
property OnAfterCellPaint: TVTAfterCellPaintEvent;
```

Description

This event is called whenever a cell has been painted. A cell is defined as being one part of a node bound to a certain column. This event is called several times per node (the amount is determined by visible columns and size of the part to draw).

See Also

Paint cycles and stages(see page 36)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.70 TBaseVirtualTree.OnAfterItemErase Event

Paint support event.

Pascal

```
property OnAfterItemErase: TVTAfterItemEraseEvent;
```

Description

Called after the background of a node has been erased (erasing can also be filling with a background image). This event is called once per node in a paint cycle.

See Also

Paint cycles and stages(see page 36)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.71 TBaseVirtualTree.OnAfterItemPaint Event

Paint support event.

Pascal

```
property OnAfterItemPaint: TVTAfterItemPaintEvent;
```

Description

Called after a node has been drawn. This event is called once per node.

See Also

Paint cycles and stages(see page 36)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.72 TBaseVirtualTree.OnAfterPaint Event

Paint support event.

Pascal

```
property OnAfterPaint: TVTPaintEvent;
```

Description

Called after all nodes which needed an update have been drawn. This event is called once per paint cycle.

See Also

Paint cycles and stages(see page 36)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.73 TBaseVirtualTree.OnBeforeCellPaint Event

Paint support event.

Pascal

```
property OnBeforeCellPaint: TVTBeforeCellPaintEvent;
```

Description

This event is called immediately before a cell is painted.

See Also

Paint cycles and stages(see page 36)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.74 TBaseVirtualTree.OnBeforeItemErase Event

Paint support event.

Pascal

```
property OnBeforeItemErase: TVTBeforeItemEraseEvent;
```

Description

Called when the background of a node is about to be erased.

See Also

Paint cycles and stages(see page 36)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.75 TBaseVirtualTree.OnBeforeItemPaint Event

Paint support event.

Pascal

```
property OnBeforeItemPaint: TVTBeforeItemPaintEvent;
```

Description

Called after the background of a node has been drawn and just before the node itself is painted. In this event the application gets the opportunity to decide whether a node should be drawn normally or should be skipped. The application can draw the node itself if necessary or leave the node area blank.

See Also

Paint cycles and stages(see page 36)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.76 TBaseVirtualTree.OnBeforePaint Event

Paint support event.

Pascal

```
property OnBeforePaint: TVTPaintEvent;
```

Description

Called as very first event in a paint cycle. In this event has the application the opportunity to do some special preparation of the canvas onto which the tree is painted, e.g. setting a special viewport and origin or a different mapping mode.

See Also

Paint cycles and stages(see page 36)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.77 TBaseVirtualTree.OnChange Event

Navigation support event.

Pascal

```
property OnChange: TVTChangeEvent;
```

Description

Called when a node's selection state has changed.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.78 TBaseVirtualTree.OnChecked Event

Check support event.

Pascal

```
property OnChecked: TVTChangeEvent;
```

Description

Triggered when a node's check state has changed.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.79 TBaseVirtualTree.OnChecking Event

Check support event.

Pascal

```
property OnChecking: TVTCheckChangingEvent;
```

Description

Triggered when a node's check state is about to change and allows to prevent the change.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.80 TBaseVirtualTree.OnCollapsed Event

Miscellaneous event.

Pascal

```
property OnCollapsed: TVTChangeEvent;
```

Description

Triggered after a node has been collapsed, that is, its child nodes are no longer displayed.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.81 TBaseVirtualTree.OnCollapsing Event

Miscellaneous event.

Pascal

```
property OnCollapsing: TVTChangingEvent;
```

Description

Triggered when a node is about to be collapsed and allows to prevent collapsing the node by setting Allowed to false.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.82 TBaseVirtualTree.OnColumnClick Event

Header and column support event.

Pascal

```
property OnColumnClick: TVTColumnClickEvent;
```

Description

Triggered when the user released a mouse button over the same column in the client area on which the button was pressed previously.

See Also

OnHeaderClick(see TBaseVirtualTree.OnHeaderClick Event, page 143)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.83 TBaseVirtualTree.OnColumnDblClick Event

Header and column support event.

Pascal

```
property OnColumnDblClick: TVTColumnDblClickEvent;
```

Description

Same as OnColumnClick(see TBaseVirtualTree.OnColumnClick Event, page 132) but for double clicks.

See Also

OnColumnClick(see TBaseVirtualTree.OnColumnClick Event, page 132), OnHeaderDblClick(see TBaseVirtualTree.OnHeaderDblClick Event, page 143)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.84 TBaseVirtualTree.OnColumnResize Event

Header and column support routine.

Pascal

```
property OnColumnResize: TVTHeaderNotifyEvent;
```

Description

Triggered when a column is being resized. During resize OnColumnResize is frequently hence you should make any code in the associated event handle a short and fast as possible.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.85 TBaseVirtualTree.OnCompareNodes Event

Sort and search support event.

Pascal

```
property OnCompareNodes: TVTCompareEvent;
```

Description

This event is the core event for all comparations between nodes. It is important that you write a handler for this event if you want to sort nodes!

Result must be set to less than 0 if Node1 is considered as being before Node2, equal to 0 if both a considered being the same and greater than 0 if the first node is considered as being after node 2. Keep in mind that you don't need to take sort direction into account. This is automatically handled by the tree. Simply return a comparation result as would there be an ascending sort order.

Below is some sample code taken from the Advanced Demo:

```
procedure TMainForm.VDT1CompareNodes(Sender: TBaseVirtualTree; Node1, Node2: PVirtualNode; Column:
Integer;
  var Result: Integer);

// used to sort the image draw tree

var
  Data1,
```

```
    Data2: PImageData;

  begin
    Data1 := Sender.GetNodeData(Node1);
    Data2 := Sender.GetNodeData(Node2);
    // folder are always before files
    if Data1.IsFolder <> Data2.IsFolder then
    begin
      // one of both is a folder the other a file
      if Data1.IsFolder then
        Result := -1
      else
        Result := 1;
    end
    else // both are of same type (folder or file)
      Result := CompareText(Data1.FullPath, Data2.FullPath);
  end;
```

See Also

SortTree(see TBaseVirtualTree.SortTree Method , page 234), Sort

Class

TBaseVirtualTree Class(see page 88)

10.1.2.86 TBaseVirtualTree.OnCreateDataObject Event

Drag'n drop support event.

Pascal

```
property OnCreateDataObject: TVTCreateDataObjectEvent;
```

Description

This event is called when the tree's drag manager needs a data object interface to start a drag'n drop operation. Descentants (which override DoGetDataObject) or the application can return an own IDataObject implementation to support special formats.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.87 TBaseVirtualTree.OnCreateDragManager Event

Drag'n drop support event.

Pascal

```
property OnCreateDragManager: TVTCreateDragManagerEvent;
```

Description

This event is usually not used but allows power users to create(see TBaseVirtualTree.Create Constructor , page 170) their own drag manager to have different actions and/or formats than the internal drag manager.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.88 TBaseVirtualTree.OnCreateEditor Event

Editing support event.

Pascal

```
property OnCreateEditor: TVTCreateEditorEvent;
```

Description

Allows to supply a customized node editor without changing the tree. TBaseVirtualTree(see TBaseVirtualTree Class, page 88) triggers this event and raises an exception if there no editor is returned. If you don't want this then disable edit support for nodes in TreeOptions.MiscOptions. Descentants like TCustomVirtualStringTree(see TCustomVirtualStringTree Class, page 274) supply a generic and simple string editor.

See Also

Editors and editing(see page 41)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.89 TBaseVirtualTree.OnDragAllowed Event

Drag'n drop support event.

Pascal

```
property OnDragAllowed: TVTDragAllowedEvent;
```

Description

This event is called in the mouse button down handler to determine whether the application allows to start a drag operation. Since this check is done in sync with the other code it is much prefered over doing a manual BeginDrag(see TBaseVirtualTree.BeginDrag Method , page 164).

Notes

The OnDragAllowed event is called only if the current DragMode is dmManual.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.90 TBaseVirtualTree.OnDragDrop Event

Drag'n drop support event.

Pascal

```
property OnDragDrop: TVTDragDropEvent;
```

Description

Triggered when either a VCL or a OLE drop action occured. Accepting drag and drop actions is not trivial. In order to maintain a minimum compatibility with the VCL drag'n drop system Virtual Tree accepts not only OLE drop actions but also those issued by the Delphi VCL (which is totally different to the OLE way, unfortunately), provided toAcceptOLEDrop is set in TreeOptions.MiscOptions. The code snippet below is taken from a sample project provided with Virtual Tree. It shows a general way to deal with dropped data. The following check list can be used as orientation and additional comment to the code:

1. Determine what kind of drop data is passed. If DataObject is nil or Formats is empty then the drag source is a VCL control. The event is not triggered for OLE drag'n drop if there is no OLE format is available (which should never occur).

2. If the event is triggered by a VCL control then use Source to access either the control or the drag object, depending on the circumstances of the action.
3. For OLE drag'n drop iterate through the Formats list to find a format you can handle.
4. If you find CF_VIRTUALTREE(see CF_VIRTUALTREE Variable, page 658) then the source of the drag operation is a Virtual Treeview. Since this is the native tree format you can pass it to the Sender's ProcessDrop(see TBaseVirtualTree.ProcessDrop Method , page 228) method which will take care to retrieve the data and act depending on Effect and Mode. No further action by the application is usually required in this case.
5. If you do not find CF_VIRTUALTREE(see CF_VIRTUALTREE Variable, page 658) then the operation has been initiated by another application, e.g. the Explorer (then you will find CF_HDROP or CF_SHELLIDLIST in formats) or Notepad (then you will get CF_TEXT and perhaps CF_UNICODETEXT) etc., depending on the data which is actually dropped.
6. Use the provided DataObject to get the drop data via IDataObject.GetData and act depending on the format you get.
7. Finally set Effect to either DROPEFFECT_COPY, DROPEFFECT_MOVE or DROPEFFECT_NONE to indicate which operation needs to be finished in Sender when the event returns. If you return DROPEFFECT_MOVE then all marked nodes in the source tree will be deleted, otherwise they stay where they are.

```
procedure TMainForm.VTDragDrop(Sender: TBaseVirtualTree; Source: TObject; DataObject: IDataObject;
  const Formats: array of Word; Shift: TShiftState; Pt: TPoint; var Effect: Integer; Mode: TDropMode);

var
  I: Integer;
  AttachMode: TVTNodeAttachMode;

begin
  if Length(Formats) > 0 then
  begin
    // OLE drag'n drop
    // If the native tree format is listed then use this and accept the drop, otherwise recject
(ignore) it.
    // It is recommend by Microsoft to order available clipboard formats in decreasing detail richness
so
    // the first best format which we can accept is usually the best format we can get at all.
    for I := 0 to High(Formats) do
      if Formats[I] = CF_VIRTUALTREE then
      begin
        case Mode of
          dmAbove:
            AttachMode := amInsertBefore;
          dmOnNode:
            AttachMode := amAddChildLast;
          dmBelow:
            AttachMode := amInsertAfter;
        else
          if Assigned(Source) and (Source is TBaseVirtualTree) and (Sender <> Source) then
            AttachMode := amInsertBefore
          else
            AttachMode := amNowhere;
        end;
        // in the case the drop target does an optimized move Effect is set to DROPEFFECT_NONE
        // to indicate this also to the drag source (so the source doesn't need to take any further
action)
        Sender.ProcessDrop(DataObject, Sender.DropTargetNode, Effect, AttachMode);
        Break;
      end;
  end
  else
  begin
    // VCL drag'n drop, Effects contains by default both move and copy effect suggestion,
    // as usual the application has to find out what operation is finally to do
    Beep;
  end;
end;
```

Class

TBaseVirtualTree Class(see page 88)

10.1.2.91 TBaseVirtualTree.OnDragOver Event

Drag'n drop support event.

Pascal

```
property OnDragOver: TVTDragOverEvent;
```

Description

Triggered when Sender is the potential target of a drag'n drop operation. You can use this event to allow or deny a drop operation by setting Allowed to True or False, respectively. For conditions of OLE or VCL drag source see OnDragDrop(see TBaseVirtualTree.OnDragDrop Event, page 135).

See Also

OnDragDrop(see TBaseVirtualTree.OnDragDrop Event, page 135)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.92 TBaseVirtualTree.OnEditCancelled Event

Editing support event.

Pascal

```
property OnEditCancelled: TVTEditCancelEvent;
```

Description

Triggered when an edit action has been cancelled.

See Also

Editors and editing(see page 41)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.93 TBaseVirtualTree.OnEdited Event

Editing support event.

Pascal

```
property OnEdited: TVTEditChangeEvent;
```

Description

Triggered when an edit action has successfully been finished.

See Also

Editors and editing(see page 41)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.94 TBaseVirtualTree.OnEditing Event

Editing support event.

Pascal

```
property OnEditing: TVTEditChangingEvent;
```

Description

Triggered when a node is about to be edited. Use Allowed to allow or deny this action.

See Also

Editors and editing(see page 41)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.95 TBaseVirtualTree.OnExpanded Event

Misscellaneous event.

Pascal

```
property OnExpanded: TVTChangeEvent;
```

Description

Triggered after a node has been expanded.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.96 TBaseVirtualTree.OnExpanding Event

Miscellaneous event.

Pascal

```
property OnExpanding: TVTChangingEvent;
```

Description

Triggered just before a node is expanded. Use Allowed to allow or deny this action.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.97 TBaseVirtualTree.OnFocusChanged Event

Navigation support event.

Pascal

```
property OnFocusChanged: TVTFocusChangeEvent;
```

Description

Triggered after the focused node changed. When examining Node keep in mind that it can be nil, meaning there is no focused node.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.98 TBaseVirtualTree.OnFocusChanging Event

Navigation support event.

Pascal

```
property OnFocusChanging: TVTFocusChangingEvent;
```

Description

Triggered when the node focus is about to change. You can use Allowed to allow or deny a focus change. Keep in mind that either the old or the new node can be nil.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.99 TBaseVirtualTree.OnFreeNode Event

Data management node.

Pascal

```
property OnFreeNode: TVTFreeNodeEvent;
```

Description

Triggered when a node is about to be freed. This is the ideal place to free/disconnect your own data you associated with Node. Keep in mind, that data which is stored directly in the node does not need to be free by the application. This is part of the node record and will be freed when the node is freed. You should however finalize the data in such a case if it contains references to external memory objects (e.g. variants, strings, interfaces).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.100 TBaseVirtualTree.OnGetCellIsEmpty Event

Triggered when the tree control needs to know whether a given column is empty.

Pascal

```
property OnGetCellIsEmpty: TVTGetCellIsEmptyEvent;
```

Description

Virtual Treeview supports the concept of column spanning where one cell with too much text to fit into its own space can expand to the right cell neighbors if they are empty. To make this work it is necessary to know if a cell is considered as being empty, whatever this means to an application. The string tree descendant simply checks the text for the given cell and calls back its ancestor if there is no text to further refine if the cell must stay as if it contained something. The ancestor (TBaseVirtualTree(see TBaseVirtualTree Class, page 88)) now triggers OnGetCellIsEmpty to let the application decide.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.101 TBaseVirtualTree.OnGetCursor Event

Miscellaneous event.

Pascal

```
property OnGetCursor: TVTGetCursorEvent;
```

Description

This event is triggered from the WM_SETCURSOR message to allow the application use several individual cursors for a tree. The Cursor property allows to set one cursor for the whole control but not to use separate cursors for different tree parts.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.102 TBaseVirtualTree.OnGetHeaderCursor Event

Header and column support event.

Pascal

```
property OnGetHeaderCursor: TVTGetHeaderCursorEvent;
```

Description

This event is triggered from the WM_SETCURSOR message to allow the application to define individual cursors for the header part of the tree control.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.103 TBaseVirtualTree.OnGetHelpContext Event

Miscellaneous event.

Pascal

```
property OnGetHelpContext: TVTHelpContextEvent;
```

Description

This event is usually triggered when the user pressed F1 while the tree has the focus. The tree is iteratively traversed all the way up to the top level parent of the given node until a valid help context index is returned (via this event). When the loop reaches the top level without getting a help index then the tree control's help index is used. If the tree itself does not have a help context index then a further traversal is initiated going up parent by parent of each control in the current window hierarchy until either a valid index is found or there is no more window parent.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.104 TBaseVirtualTree.OnGetImageIndex Event

Display management event.

Pascal

```
property OnGetImageIndex: TVTGetImageEvent;
```

Description

This event is triggered whenever the tree needs the index of an image, be it the normal, the selected or the state image. The event should be as fast as possible because it is at times frequently called when the layout of the node must be determined, e.g. while doing draw selection with the mouse or painting the tree. Kind determines which image is needed and Column determines for which column of the node the image is needed. This value can be -1 to indicate there is no column used. The parameter Ghosted can be set to true to blend the image 50% against the tree background and can be used for instance in explorer trees to mark hidden file system objects. Additionally nodes are also drawn with a ghosted icon if the are part of a cut set during a pending cut-to-clipboard operation. In this case changing the ghosted parameter has no effect.

Notes

Blending nodes can be switched by using toUseBlendImages in TreeOptions.PaintOptions.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.105 TBaseVirtualTree.OnGetImageIndexEx Event

Not documented.

Pascal

```
property OnGetImageIndexEx: TVTGetImageExEvent;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.106 TBaseVirtualTree.OnGetLineStyle Event

Display management event.

Pascal

```
property OnGetLineStyle: TVTGetLineStyleEvent;
```

Description

This event is used to customize the appearance of the tree and grid lines and is only triggered if the LineStyle(see TBaseVirtualTree.LineStyle Property, page 125) property is set to lsCustomStyle. The event must return a pointer to an array containing bits for an 8 x 8 pixel image with word aligned entries. For more info see PrepareBitmaps and the Windows APIs CreateBitmap and CreatePatternBrush.

Notes

It is important that you do not use dynamically allocated memory in this event (also no local variables on the stack). If you do so then either the memory is not valid on return of the event (if allocated on stack) or will never be freed (if allocated with a memory manager). Instead use a constant array and return its address.

See Also

PrepareBitmaps

Class

TBaseVirtualTree Class(🡅 see page 88)

10.1.2.107 TBaseVirtualTree.OnGetNodeDataSize Event

Data management event.

Pascal

```
property OnGetNodeDataSize: TVTGetNodeDataSizeEvent;
```

Description

Triggered when access to a node's data happens the first time but the actual data size is not yet set. Usually you would specify the size of the data you want to have added to each node by NodeDataSize(🡅 see TBaseVirtualTree.NodeDataSize Property, page 127), e.g. SizeOf(TMyRecord) is quite usual there (where TMyRecord is the structure you want to have stored in the node). Sometimes, however it is not possible to determine the node size in advance, so you can leave NodeDataSize(🡅 see TBaseVirtualTree.NodeDataSize Property, page 127) being -1 (the default value) and the OnGetNodeDataSize event is triggered as soon as the first regular node is created (the hidden root node does not have user data but internal data which is determined by other means).

See Also

NodeDataSize(🡅 see TBaseVirtualTree.NodeDataSize Property, page 127), Data handling(🡅 see page 39)

Class

TBaseVirtualTree Class(🡅 see page 88)

10.1.2.108 TBaseVirtualTree.OnGetPopupMenu Event

Miscellaneous event.

Pascal

```
property OnGetPopupMenu: TVTPopupEvent;
```

Description

This event allows the application to return a popup menu which is specific to a certain node. The tree does an automatic traversal all the way up to the top level node which is the parent of a given node to get a popup menu. If Menu is set then the traversal stops. Otherwise it continues until either a menu is set, AskParent is set to False or the top level parent has been reached.

Class

TBaseVirtualTree Class(🡅 see page 88)

10.1.2.109 TBaseVirtualTree.OnGetUserClipboardFormats Event

Drag'n drop and clipboard support event.

Pascal

```
property OnGetUserClipboardFormats: TVTGetUserClipboardFormatsEvent;
```

Description

Whenever the tree needs to specify the available clipboard formats for a clipboard or drag'n drop operation it calls this event too, to allow the application or descentants (which would override DoGetUserClipboardFormats(🡅 see

TBaseVirtualTree.DoGetUserClipboardFormats Method , page 187)) to specify own formats which can be rendered. Since the build-in data object does not know how to render formats which are specified here you have to supply a handler for the OnRenderOLEData(see TBaseVirtualTree.OnRenderOLEData Event, page 150) event or an own IDataObject implementation to fully support your own formats.

Use the Formats parameter which is an open array and add the identifiers of your formats (which you got when you registered the format).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.110 TBaseVirtualTree.OnHeaderClick Event

Header & column support event.

Pascal

```
property OnHeaderClick: TVTHeaderClickEvent;
```

Description

This event is triggered when the user clicks on a header button and is usually a good place to set the current SortColumn and SortDirection.

See Also

SortColumn, SortDirection

Class

TBaseVirtualTree Class(see page 88)

10.1.2.111 TBaseVirtualTree.OnHeaderDblClick Event

Header & column support event.

Pascal

```
property OnHeaderDblClick: TVTHeaderClickEvent;
```

Description

Unlike OnHeaderClick(see TBaseVirtualTree.OnHeaderClick Event, page 143) this event is triggered for double clicks on any part of the header and comes with more detailed information like shift state, which mouse button caused the event and the mouse position.

See Also

OnHeaderClick(see TBaseVirtualTree.OnHeaderClick Event, page 143)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.112 TBaseVirtualTree.OnHeaderDragged Event

Header & column support event.

Pascal

```
property OnHeaderDragged: TVTHeaderDraggedEvent;
```

Description

Triggered after the user has released the left mouse button when a header drag operation was active. Column contains the index of the column which was dragged. Use this index for the Columns property of the header to find out the current position. OldPosition is the position which Column occupied before it was dragged around.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.113 TBaseVirtualTree.OnHeaderDraggedOut Event

Header & column support event.

Pascal

```
property OnHeaderDraggedOut: TVTHeaderDraggedOutEvent;
```

Description

When during a header drag operation the mouse moves out of the header rectangle and the mouse button is released then an OnHeaderDraggedOut event will be fired with the target mouse position in screen coordinates.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.114 TBaseVirtualTree.OnHeaderDragging Event

Header & column support event.

Pascal

```
property OnHeaderDragging: TVTHeaderDraggingEvent;
```

Description

Triggered just before dragging(see TBaseVirtualTree.Dragging Method , page 198) of a header button starts. Set Allowed to False if you want to prevent the drag operation of the given column.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.115 TBaseVirtualTree.OnHeaderDraw Event

Header & column support event.

Pascal

```
property OnHeaderDraw: TVTHeaderPaintEvent;
```

Description

If you set the hoOwnerDraw style in TVTHeader.Options(see TVTHeader.Options Property, page 548) and a column has been set to vsOwnerDraw (see also TVirtualTreeColumn.Style(see TVirtualTreeColumn.Style Property, page 490)) then OnDrawHeader is called whenever a column needs painting.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.116 TBaseVirtualTree.OnHeaderDrawQueryElements Event

Header & column support event.

Pascal

```
property OnHeaderDrawQueryElements: TVTHeaderPaintQueryElementsEvent;
```

Description

Used for advanced header painting to query the application for the elements, which are drawn by it and which should be drawn by the tree.

See Also

OnAdvancedHeaderDraw(🏠 see TBaseVirtualTree.OnAdvancedHeaderDraw Event, page 128)

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.117 TBaseVirtualTree.OnHeaderMouseDown Event

Header & column support event.

Pascal

```
property OnHeaderMouseDown: TVTHeaderMouseEvent;
```

Description

This event is similar to OnHeaderClick(🏠 see TBaseVirtualTree.OnHeaderClick Event, page 143) but comes with more detailed information like shift state, which mouse button caused the event and the mouse position.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.118 TBaseVirtualTree.OnHeaderMouseMove Event

Header & column support event.

Pascal

```
property OnHeaderMouseMove: TVTHeaderMouseMoveEvent;
```

Description

This event is triggered when the mouse pointer is moved over the header area.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.119 TBaseVirtualTree.OnHeaderMouseUp Event

Header & column support event.

Pascal

```
property OnHeaderMouseUp: TVTHeaderMouseEvent;
```

Description

This event is very much like OnHeaderMouseDown(see TBaseVirtualTree.OnHeaderMouseDown Event, page 145) but is triggered when a mouse button is released.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.120 TBaseVirtualTree.OnHotChange Event

Navigation support event.

Pascal

```
property OnHotChange: TVTHotNodeChangeEvent;
```

Description

This event is triggered if hot tracking is enabled (see also TreeOptions.PaintOptions) and when the mouse pointer moves from one node caption to another. In full row select mode most parts of a node are considered as being part of the caption.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.121 TBaseVirtualTree.OnIncrementalSearch Event

Miscellaneous event.

Pascal

```
property OnIncrementalSearch: TVTIncrementalSearchEvent;
```

Description

This event is integral part of the incremental search functionality (see also Keyboard, hotkeys and incremental search). It is triggered during search for a node which matches the given string. Similar to other compare routines return a value < 0 if the node's caption is considered as being before the given text, = 0 if it is the same and > 0 if it is considered being after the given text.

```
procedure TfrmProperties.VST3IncrementalSearch(Sender: TBaseVirtualTree; Node: PVirtualNode; const
Text: WideString;
  var Result: Integer);

var
  S, PropText: string;


begin
  // Note: This code requires a proper Unicode/WideString comparison routine which I did not want to
link here for
  // size and clarity reasons. For now strings are (implicitly) converted to ANSI to make the
comparison work.
  // Search is not case sensitive.
  S := Text;
  if Node.Parent = Sender.RootNode then
  begin
    // root nodes
    if Node.Index = 0 then
```

```
        PropText := 'Description'
      else
        PropText := 'Origin';
    end
    else
    begin
      PropText := PropertyTexts[Node.Parent.Index, Node.Index, ptkText];
    end;

    // By using StrLIComp we can specify a maximum length to compare. This allows us to find also nodes
    // which match only partially.
    Result := StrLIComp(PChar(S), PChar(PropText), Min(Length(S), Length(PropText)))
  end;
```

Notes

Usually incremental search allows to match also partially. Hence it is recommended to do comparison only up to the length

of the shorter string.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.122 TBaseVirtualTree.OnInitChildren Event

Node management event.

Pascal

```
property OnInitChildren: TVTInitChildrenEvent;
```

Description

In order to allow the tree only to fill content where needed it is possible to set the vsHasChildren style in a node's initializaton whithout really adding any child nodes. These child nodes must be initialized first when they are about to be displayed or another access (like search, iteration etc.) occurs.

The application usually prepares data needed to fill child nodes when they are initialized and retrieves the actual number. Set ChildCount(see TBaseVirtualTree.ChildCount Property, page 111) to the number of children you want.

See Also

The virtual paradigm(see page 33)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.123 TBaseVirtualTree.OnInitNode Event

Node management event.

Pascal

```
property OnInitNode: TVTInitNodeEvent;
```

Description

This event is important to connect the tree to your internal data. It is the ideal place to put references or whatever you need into a node's data area. You can set some initial states like selection, expansion state or that a node has child nodes.

See Also

The virtual paradigm(see page 33)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.124 TBaseVirtualTree.OnKeyAction Event

Miscellaneous event.

Pascal

```
property OnKeyAction: TVTKeyActionEvent;
```

Description

This event is a convinient way for the application or descentant trees to change the semantic of a certain key stroke. It is triggered when the user presses a key and allows either to process that key normally (leave DoDefault being True) or change it to another key instead (set DoDefault to False then). This way a key press can change its meaning or entirely be ignored (if CharCode is set to 0).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.125 TBaseVirtualTree.OnLoadNode Event

Streaming support event.

Pascal

```
property OnLoadNode: TVTSaveNodeEvent;
```

Description

This event is typically triggered when serialized tree data must be restored, e.g. when loading the tree from file or stream or during a clipboard/drag'n drop operation. You should only read in what you wrote out in OnSaveNode(see TBaseVirtualTree.OnSaveNode Event, page 151). For safety there is a check in the loader code which tries to keep the internal serialization structure intact in case the application does not read correctly.

See Also

OnSaveNode(see TBaseVirtualTree.OnSaveNode Event, page 151), LoadFromStream(see TBaseVirtualTree.LoadFromFile Method , page 223), SaveToStream(see TBaseVirtualTree.SaveToFile Method , page 232), AddFromStream(see TBaseVirtualTree.AddFromStream Method , page 161), VTTreeStreamVersion(see VTTreeStreamVersion Constant, page 695), TVTHeader.LoadFromStream(see TVTHeader.LoadFromStream Method , page 554), TVTHeader.SaveToStream(see TVTHeader.SaveToStream Method , page 556)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.126 TBaseVirtualTree.OnMeasureItem Event

Miscellaneous event.

Pascal

```
property OnMeasureItem: TVTMeasureItemEvent;
```

Description

Virtual Treeview supports individual node heights. However it might sometimes unpractical to set this height in advance (e.g. during OnInitNode(🏠 see TBaseVirtualTree.OnInitNode Event, page 147)). Another scenario might be that multi line nodes must size themselves to accomodate the entire node text without clipping. For such and similar cases the event OnMeasureItem is for. It is queried once for each node and allows to specify the node's future height. If you later want to have a new height applied (e.g. because the node's text changed) then call InvalidateNode(🏠 see TBaseVirtualTree.InvalidateNode Method , page 221) for it and its vsHeightMeasured state is reset causing so the tree to trigger the OnMeasureItem event again when the node is painted the next time.

See Also

InvalidateNode(🏠 see TBaseVirtualTree.InvalidateNode Method , page 221), vsHeightMeasured

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.127 TBaseVirtualTree.OnNodeCopied Event

Miscellaneous event.

Pascal

```
property OnNodeCopied: TVTNodeCopiedEvent;
```

Description

This event is triggered during drag'n drop after a node has been copied to a new location. Sender is the target tree where the copy operation took place.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.128 TBaseVirtualTree.OnNodeCopying Event

Miscellaneous event.

Pascal

```
property OnNodeCopying: TVTNodeCopyingEvent;
```

Description

This event is triggered when a node is about to be copied to a new location. Use Allowed to allow or deny the action. Sender is the target tree where the copy operation will take place.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.129 TBaseVirtualTree.OnNodeMoved Event

Miscellaneous event.

Pascal

```
property OnNodeMoved: TVTNodeMovedEvent;
```

Description

This event is very much like OnNodeCopied(🏠 see TBaseVirtualTree.OnNodeCopied Event, page 149) but used for moving

nodes instead.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.130 TBaseVirtualTree.OnNodeMoving Event

Miscellaneous event.

Pascal

```
property OnNodeMoving: TVTNodeMovingEvent;
```

Description

This event is very much like OnNodeCopying(see TBaseVirtualTree.OnNodeCopying Event, page 149) but used for moving nodes instead.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.131 TBaseVirtualTree.OnPaintBackground Event

Paint support event.

Pascal

```
property OnPaintBackground: TVTBackgroundPaintEvent;
```

Description

This event is triggered when the tree has finished its painting and there is an area which is not covered by nodes. For nodes there are various events to allow background customizaton. For the free area in the tree window there is this event.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.132 TBaseVirtualTree.OnRenderOLEData Event

Drag'n drop and clipboard support event.

Pascal

```
property OnRenderOLEData: TVTRenderOLEDataEvent;
```

Description

This event is triggered when the data in a clipboard or drag'n drop operation must be rendered but the built-in data object does not know the requested format. This is usually the case when the application (or descentants) have specified their own formats in OnGetUserClipboardFormats(see TBaseVirtualTree.OnGetUserClipboardFormats Event, page 142).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.133 TBaseVirtualTree.OnResetNode Event

Node management event.

Pascal

```
property OnResetNode: TVTChangeEvent;
```

Description

For large trees or simply because the content changed it is sometimes necessary to discard a certain node and release all its children. This can be done with ResetNode(see TBaseVirtualTree.ResetNode Method , page 231) which will trigger this event.

See Also

ResetNode(see TBaseVirtualTree.ResetNode Method , page 231)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.134 TBaseVirtualTree.OnSaveNode Event

Streaming support event.

Pascal

```
property OnSaveNode: TVTSaveNodeEvent;
```

Description

This event is triggered whenever a certain node must be serialized into a stream, e.g. for saving to file or for copying to another tree/node during a clipboard or drag'n drop operation. Make sure you only store non-transient data into the stream. Pointers (including long/wide string references) are transient and the application cannot assume to find the data a pointer references on saving at the same place when the node is loaded(see TBaseVirtualTree.Loaded Method , page 223) (see also OnLoadNode(see TBaseVirtualTree.OnLoadNode Event, page 148)). This is even more essential for nodes which are moved or copied between different trees in different processes (applications). Storing strings however is easily done by writing the strings as a whole into the stream.

Notes

For exchanging data between different trees and for general stability improvement I strongly recommend that you insert a kind of identifier as first stream entry when saving a node. This identifier can then be used to determine what data will follow when loading the node later and does normally not required to be stored in the node data.

See Also

OnLoadNode(see TBaseVirtualTree.OnLoadNode Event, page 148), LoadFromStream(see TBaseVirtualTree.LoadFromFile Method , page 223), SaveToStream(see TBaseVirtualTree.SaveToFile Method , page 232), AddFromStream(see TBaseVirtualTree.AddFromStream Method , page 161), VTTreeStreamVersion(see VTTreeStreamVersion Constant, page 695), TVTHeader.LoadFromStream(see TVTHeader.LoadFromStream Method , page 554), TVTHeader.SaveToStream(see TVTHeader.SaveToStream Method , page 556)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.135 TBaseVirtualTree.OnScroll Event

Miscellaneous event.

Pascal

```
property OnScroll: TVTScrollEvent;
```

Description

This event is triggered when the tree is scrolled horizontally or vertically. You can use it to synchronize scrolling of several trees or other controls.

See Also

OffsetXY(see TBaseVirtualTree.OffsetXY Property, page 128)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.136 TBaseVirtualTree.OnShowScrollbar Event

Not documented.

Pascal

```
property OnShowScrollbar: TVTScrollbarShowEvent;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.137 TBaseVirtualTree.OnStateChange Event

Miscellaneous event.

Pascal

```
property OnStateChange: TVTStateChangeEvent;
```

Description

For special effects or in order to increase performance it is sometimes useful to know when the tree changes one of its internal states like tsIncrementalSearching or tsOLEDragging. The OnStateChange event is triggered each time such a change occurs letting so the application take measures for it.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.138 TBaseVirtualTree.OnStructureChange Event

Miscellaneous event.

Pascal

```
property OnStructureChange: TVTStructureChangeEvent;
```

Description

This event is triggered when a change in the tree structure is made. That means whenever a node is created or destroyed or a node's child list is change (because a child node was moved, copied etc.) then OnStructureChange is executed.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.139 TBaseVirtualTree.OnUpdating Event

Miscellaneous event.

Pascal

```
property OnUpdating: TVTUpdatingEvent;
```

Description

This event is triggered when the application or the tree call BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164) or EndUpdate(see TBaseVirtualTree.EndUpdate Method , page 201) and indicate so when a larger update operation takes place. This can for instance be used to show a hour glass wait cursor.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.140 TBaseVirtualTree.RootNode Property

Reference to the internal root node which is the anchor of the entire tree node hierarchy.

Pascal

```
property RootNode: PVirtualNode;
```

Description

For anchoring the tree hierarchy an internal tree node is maintained which is mostly just like any other tree node but has sometimes differently handled. The root node is always expanded and initialized. Its parent member points to the treeview to which the node belongs to and its PreviousSibling and NextSibling members point to the root node itself to make it possible to actually recognize this node.

Notes

You should not use the root node to iterate through the tree. It is only publicly accessible because it is the parent of

all top level nodes and can be used to test a node whether it is a top level node or not.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.141 TBaseVirtualTree.RootNodeCount Property

Read or set the number of nodes on the top level.

Pascal

```
property RootNodeCount: Cardinal;
```

Description

Usually setting RootNodeCount is all what is needed to initially fill the tree. When one of the top level nodes is initialized

you can set its ivsHasChildren style. This will then cause to ask to initialize the child nodes. Recursively applied, you can use this principle to create(see TBaseVirtualTree.Create Constructor , page 170) tree nodes on demand (e.g. when their parent is expanded).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.142 TBaseVirtualTree.ScrollBarOptions Property

Reference to the scroll bar options class.

Pascal

```
property ScrollBarOptions: TScrollBarOptions;
```

Description

Like many other aspects in Virtual Treeview also scrollbars can be customized. See the class itself for further descriptions.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.143 TBaseVirtualTree.SearchBuffer Property

Current input string for incremental search.

Pascal

```
property SearchBuffer: WideString;
```

Description

When incremental search is active you can use SearchBuffer to get the input string typed by the user, which created the last match.

See Also

IncrementalSearch(see TBaseVirtualTree.IncrementalSearch Property, page 122)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.144 TBaseVirtualTree.Selected Property

Property to modify or determine the selection state of a node.

Pascal

```
property Selected [Node: PVirtualNode]: Boolean;
```

Description

This array property is used to test whether a given node is selected or to switch its selection state. Note that the selection state has nothing to do with the focused state(see TBaseVirtualTree.FocusedNode Property, page 118). Only one node can be focused while any number of nodes can be selected (read: can be marked with the selection flag to paint their caption differently). Selection is mainly used to mark nodes for clipboard and drag'n drop operations.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.145 TBaseVirtualTree.SelectedCount Property

Contains the number of selected nodes.

Pascal

```
property SelectedCount: Integer;
```

Description

If multiselection is enabled (toMultiSelect) then SelectedCount will contain the actual number of selected nodes. In order to change the selection state of a node use Selected(see TBaseVirtualTree.Selected Property, page 154) or AddToSelection(see TBaseVirtualTree.AddToSelection Method (PVirtualNode), page 161)/RemoveFromSelection(see TBaseVirtualTree.RemoveFromSelection Method , page 230).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.146 TBaseVirtualTree.SelectionBlendFactor Property

Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

Pascal

```
property SelectionBlendFactor: Byte;
```

Description

For a visually appealing tree some operations use alpha blending. One of these operations is multi selection using the mouse. Another one is the rectangle drawn around the caption of selected nodes. Both rectangles use the SelectionBlendFactor to determine how much of the underlying tree image and how much of the rectangles should be seen. The factor can be in the range of [0..255] where 0 means the rectangle is fully transparent and 255 it is fully opaque.

If you don't like to use blended node selection rectangles then switch them off by removing toUseBlendedSelection from TVTPaintOptions(see TVTPaintOptions Type, page 625). For selecting a certain multi selection rectangle style use DrawSelectionMode(see TBaseVirtualTree.DrawSelectionMode Property, page 116).

Notes

Alpha blending is only enabled when the current processor supports MMX instructions. If MMX is not supported then a dotted draw selection rectangle and an opaque node selection rectangle is used.

See Also

DrawSelectionMode(see TBaseVirtualTree.DrawSelectionMode Property, page 116), TVTPaintOptions(see TVTPaintOptions Type, page 625)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.147 TBaseVirtualTree.SelectionCurveRadius Property

Read or set the current corner radius for node selection rectangles.

Pascal

```
property SelectionCurveRadius: Cardinal;
```

Description

This is a special property to determine the radius of the corners of the selection rectangle for a node caption. Virtual Treeview supports not only simple rectangular selection marks but also such with rounded corners. This feature, however, is only available if blended node selection rectangles are disabled.

See Also

SelectionBlendFactor(see TBaseVirtualTree.SelectionBlendFactor Property, page 155), DrawSelectionMode(see TBaseVirtualTree.DrawSelectionMode Property, page 116), TVTPaintOptions(see TVTPaintOptions Type, page 625)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.148 TBaseVirtualTree.StateImages Property

Reference to the images list which is used for the state images.

Pascal

```
property StateImages: TCustomImageList;
```

Description

Each node can (in each column) have several images. One is the check image which is supplied by internal image lists or a special external list (see also CustomCheckImages(see TBaseVirtualTree.CustomCheckImages Property, page 113)). Another one is the state image and yet another one the normal/selected image.

See Also

CheckImages(see TBaseVirtualTree.CheckImages Property, page 111), Images(see TBaseVirtualTree.Images Property, page 122)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.149 TBaseVirtualTree.TextMargin Property

Read or set the distance of the node caption to its borders.

Pascal

```
property TextMargin: Integer;
```

Description

TextMargin is used to define a border like area within the content rectangle of a node. This rectangle is the area of the node less the space used for indentation, images, lines and node margins and usually contains the text of a node. In order to support finer adjustment there is another margin, which only applies to the left and right border in the content rectangle. This is the text margin.

See Also

Margin(see TBaseVirtualTree.Margin Property, page 125)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.150 TBaseVirtualTree.TopNode Property

The top node is the node which is currently at the top border of the client area.

Pascal

```
property TopNode: PVirtualNode;
```

Description

This property is a reference to the node which is the first node which is at least partially visible in the client area.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.151 TBaseVirtualTree.TotalCount Property

Returns the number of nodes in the tree.

Pascal

```
property TotalCount: Cardinal;
```

Description

Use this property to get the overall number of nodes currently in the tree. This will validate all nodes in the control so that also not yet created child nodes are counted.

Notes

This property is quite counter productive as it causes the entire tree to be validated when queried. This means that each node is initialized, including its children and grandchildren etc. creating so a full blown treeview (if not already done) which might keep much memory allocated (not counted the time necessary to validate all nodes). Therefore I discourage the use of the property unless it is really necessary.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.152 TBaseVirtualTree.TotalInternalDataSize Property

Keeps the currently accumulated data size for one node.

Pascal

```
property TotalInternalDataSize: Cardinal;
```

Description

Each node in the tree not only supports user data but also an interal area where TVirtualBaseTree descentants can store their own data per node. This internal data area must be allocated by a tree class, that means it must register its need for internal data. The internal data size registered by each descendant is accumulated in the TotalInternalDataSize member and is used to compute the user data offset in the node record.

See Also

Data handling(see page 39)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.153 TBaseVirtualTree.TreeOptions Property

Reference to the tree's options.

Pascal

```
property TreeOptions: TCustomVirtualTreeOptions;
```

Description

The tree options are one of the main switchs to modify a treeview's behavior. Virtual Treeview supports customizing tree options by descentants. This allows very fine adjustments for derived tree classes, including the decision which properties should be published. For more information about the base options see TCustomVirtualTreeOptions(see TCustomVirtualTreeOptions Class, page 308) and its descentants.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.154 TBaseVirtualTree.TreeStates Property

Property which keeps a set of flags which indicate current operation and states of the tree.

Pascal

```
property TreeStates: TVirtualTreeStates;
```

Description

Often it is extremly helpful to know what action is currently happening in the tree. TreeStates gives you this information, be it that the caches are currently validated, a drag operation is in progress, the tree has delayed data on the clipboard or a large update operation is under work. You can greatly optimize your code with this knowledge.

See Also

OnStateChange(see TBaseVirtualTree.OnStateChange Event, page 152)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.155 TBaseVirtualTree.UpdateCount Property

Not documented.

Pascal

```
property UpdateCount: Cardinal;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.156 TBaseVirtualTree.VerticalAlignment Property

Used to set a node's vertical button aligment with regard to the entire node rectangle.

Pascal

```
property VerticalAlignment [Node: PVirtualNode]: Byte;
```

Description

The given value is interpreted differently depending on the value of NodeAlignment(see TBaseVirtualTree.NodeAlignment Property, page 126). By default the alignment used relatively with regard to the top bound. In this case a range of 0 through 100 must be used which denotes the relative pixel amount in percent. The other variants work with absolute pixel values from top or bottom bound.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.157 TBaseVirtualTree.VisibleCount Property

Number of currently visible nodes.

Pascal

```
property VisibleCount: Cardinal;
```

Description

Visible nodes are those nodes which have the vsVisible flag set in their states.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.158 TBaseVirtualTree.VisiblePath Property

Property to set or determine a node parent's expand states.

Pascal

```
property VisiblePath [Node: PVirtualNode]: Boolean;
```

Description

A node has a visible path when all of its parent nodes are expanded. Setting this property to True will expand all parent nodes of Node if not yet done.

See Also

Visible

Class

TBaseVirtualTree Class(see page 88)

10.1.2.159 TBaseVirtualTree.WantTabs Property

Read or set whether the tree wants to process tabs on its own.

Pascal

```
property WantTabs: Boolean;
```

Description

Usually tab kex strokes advance the input focus from one control to another on a form. For special processing however it is necessary to let the control decide what to do with the given tabulator character. Virtual Treeview needs this

character mainly for its grid emulation.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.160 TBaseVirtualTree.AbsoluteIndex Method

Reads the overall index of a node.

Pascal

```
function AbsoluteIndex(Node: PVirtualNode): Cardinal;
```

Description

Indicates the index of the tree node relative to the first tree node in a tree.

Notes

Similar to TotalCount(see TBaseVirtualTree.TotalCount Property, page 157) also with AbsoluteIndex the entire tree will be validated, with all consequences like high memory

usage etc. And since Virtual Treeview is a highly changing environment there is not much sense to use the absolute index.

You cannot use it in any method or property of the control.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.161 TBaseVirtualTree.AddChild Method

Creates and adds a new child node to given node.

Pascal

```
function AddChild(Parent: PVirtualNode; UserData: Pointer = nil): PVirtualNode;
```

Description

The new node will be created as last child of Parent and is returned as result.

Notes

Using AddChild is not recommended. The method is merely there for easier migration from TTreeview. The reason is that the

method has to validate the node and does some other processing, which prevents the tree from utilizings its virtual

paradigm. Important advantages will so disappear. If possible you should restructure your design and try to use the right

way: via OnInitNode(see TBaseVirtualTree.OnInitNode Event, page 147) and OnInitChildren(see TBaseVirtualTree.OnInitChildren Event, page 147).

See Also

InsertNode(see TBaseVirtualTree.InsertNode Method , page 217), OnInitNode(see TBaseVirtualTree.OnInitNode Event, page 147), OnInitChildren(see TBaseVirtualTree.OnInitChildren Event, page 147)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.162 TBaseVirtualTree.AddFromStream Method

Adds the content from the given stream to the given node.

Pascal

```
procedure AddFromStream(Stream: TStream; TargetNode: PVirtualNode);
```

Description

AddFromStream restores the subtree stored in Stream and adds it to TargetNode. The content of the stream must have been saved previously with SaveToStream(see TBaseVirtualTree.SaveToFile Method , page 232).

See Also

SaveToStream(see TBaseVirtualTree.SaveToFile Method , page 232)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.163 AddToSelection

10.1.2.163.1 TBaseVirtualTree.AddToSelection Method (PVirtualNode)

Adds one or more nodes to the current selection.

Pascal

```
procedure AddToSelection(Node: PVirtualNode); virtual; overload;
```

Description

AddToSelection either takes a single node or an array of nodes and adds them to the current seletion in the tree. In this process also the vsSelected state of the node is set. NewLength is the amount of nodes to add (necessary to allow NewItems to be larger than the actual used entries). ForceInsert is true if nodes must be inserted without consideration of level select constraint or already set selected flags (e.g. when loading from stream).

Notes

In the case ForceInsert is true the caller is responsible for making sure the new nodes aren't already in the

selection array!

Class

TBaseVirtualTree Class(see page 88)

10.1.2.163.2 TBaseVirtualTree.AddToSelection Method (TNodeArray, Integer, Boolean)

```
procedure AddToSelection(const NewItems: TNodeArray; NewLength: Integer; ForceInsert: Boolean =
False); virtual; overload;
procedure AddToSelection(const NewItems: TNodeArray; NewLength: Integer; ForceInsert: Boolean =
False); virtual; overload;
```

Class

TBaseVirtualTree Class(see page 88)

10.1.2.164 TBaseVirtualTree.AdjustPaintCellRect Method

Used in descentants to modify the clip rectangle of the current column while painting a certain node.

Pascal

```
procedure AdjustPaintCellRect(var PaintInfo: TVTPaintInfo; var NextNonEmpty: TColumnIndex); virtual;
```

Description

The rectangle for the given cell (node, column pair in PaintInfo) can be adjusted by descendants to make room for special drawings, if necessary.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.165 TBaseVirtualTree.AdjustPanningCursor Method

Loads the proper cursor which indicates into which direction scrolling is done.

Pascal

```
procedure AdjustPanningCursor(X: Integer; Y: Integer); virtual;
```

Description

Wheel mice support a special mode for their wheel, which is used in many applications. By pressing the wheel (which is also a button) you can start so called *wheell panning*. In this mode the tree window is smoothly scrolled in the direction to which the mouse pointer is moved. As soon as you release the wheel button wheel panning is stopped. A second form of this feature is referred to as *wheel scrolling*. It is basically the same as wheel panning but is entered when you release the wheel button before you moved the mouse. In this mode you can move the mouse and do the tree scrolling without holding the wheel all the time. To stop this mode simple turn the wheel, or click any mouse button. Also pressing ESC will cause to leave the wheel scrolling mode.

Depending on the direction the tree content is scroll also the mouse cursor must be adjusted to indicate this direction. AdjustPanningCursor does this.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.166 TBaseVirtualTree.AdviseChangeEvent Method

Used to register a delayed change event.

Pascal

```
procedure AdviseChangeEvent(StructureChange: Boolean; Node: PVirtualNode; Reason: TChangeReason);
virtual;
```

Description

Often there can be many change events in a row and calling the application for each of them might be too time costly. So they are by default accumulated until a certain time has elapsed (ChangeDelay(see TBaseVirtualTree.ChangeDelay Property, page 110)) or, if BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164) was called, until EndUpdate(see TBaseVirtualTree.EndUpdate Method , page 201) is executed. If StructureChange(see TBaseVirtualTree.StructureChange Method , page 235) is False then we have a selection change event (without a specific reason) otherwise it is a structure change.

There are two possibilities to avoid delayed change events. One is the permanent way by setting ChangeDelay(see TBaseVirtualTree.ChangeDelay Property, page 110) to 0, the other one is to enter the synchronous mode by calling BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.167 TBaseVirtualTree.AllocateInternalDataArea Method

Registration method to allocate tree internal data per node.

Pascal

```
function AllocateInternalDataArea(Size: Cardinal): Cardinal; virtual;
```

Description

This method is used for descentants to specify their need for internal data. Each node contains some extra reserved bytes between the node's normal members and the user data area. This internal area can be used to cache additional information, e.g. the string tree keeps here the width of the node's caption in the main column for quick hit tests when doing draw selection with the mouse.

A tree implementation must call this method only once and before any node is created (except the hidden root node which is handled accordingly). The result value is the offset from the start of the node to the internal data area of the node for this tree class. I recommend to implement an access method called InternalData(see TBaseVirtualTree.InternalData Method , page 219) (as shown in TCustomVirtualStringTree(see TCustomVirtualStringTree Class, page 274)) which does the pointer mathematic.

See Also

Data handling(see page 39), TotalInternalDataSize(see TBaseVirtualTree.TotalInternalDataSize Property, page 157)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.168 TBaseVirtualTree.Animate Method

Support method for animated actions in the tree view.

Pascal

```
procedure Animate(Steps: Cardinal; Duration: Cardinal; Callback: TVTAnimationCallback; Data: Pointer);
virtual;
```

Description

This method is a general purpose helper to do an animation and is used for hint fading, animated node toggling etc. The method automatically takes care that the animation is done within the specified time interval. For each step in the animation loop the provided callback is called which gets Data passed as parameter.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.169 TBaseVirtualTree.Assign Method

Used to copy properties from another Virtual Treeview.

Pascal

```
procedure Assign(Source: TPersistent); override;
```

Description

Although this method assignes most tree properties it does not assign the header and the nodes to the new tree. There is an own method (TVTHeader.Assign(see TVTHeader.Assign Method , page 550)) for the header assignment. In order to copy the nodes you must save them to a stream and restore them in the other control

Class

TBaseVirtualTree Class(see page 88)

10.1.2.170 TBaseVirtualTree.BeginDrag Method

Starts an OLE drag'n drop operation.

Pascal

```
procedure BeginDrag(Immediate: Boolean; Threshold: Integer = -1);
```

Description

This method is called within the mouse down handler when DragMode is set to dmAutomatic. Manual start of a drag operation is not recommended as it confuses the correct mouse down handling which is quite complex in Virtual Treevew. If you selectively want to allow to start a drag operation then use the OnDragAllowed(see TBaseVirtualTree.OnDragAllowed Event, page 135) event which is called when DragMode is dmManual.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.171 TBaseVirtualTree.BeginSynch Method

Enters the tree into a special synchronized mode.

Pascal

```
procedure BeginSynch;
```

Description

Similar to BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164) does BeginSynch provide a mechanism to bring certain events into a common line. That means, whenever you need to make sure change events are called before a modification in the tree is finished (e.g. when changing the focus or selection) then use the synchronous mode started with BeginSynch (and stopped with EndSynch(see TBaseVirtualTree.EndSynch Method , page 200)).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.172 TBaseVirtualTree.BeginUpdate Method

Locks the tree view to perform several update operations.

Pascal

```
procedure BeginUpdate;
```

Description

Call this method when a long lasting operation begins which might involve manipulation of many nodes.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.173 TBaseVirtualTree.CalculateSelectionRect Method

Support method for draw selection.

Pascal

```
function CalculateSelectionRect(X: Integer; Y: Integer): Boolean; virtual;
```

Description

Recalculates old and new selection rectangle given that X, Y are new mouse coordinates. The function returns true if there was a change since the last call.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.174 TBaseVirtualTree.CanAutoScroll Method

Determines whether the tree can currently auto scroll its window.

Pascal

```
function CanAutoScroll: Boolean; virtual;
```

Description

This method was created because the conditions when the tree may automatically scroll its content are quite complex. Additionally, tree descendants might want to add further limitations. Thus the determination has been put into an own method which returns true if the tree is allowed to scroll, otherwise False.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.175 TBaseVirtualTree.CancelCutOrCopy Method

Canceles any pending cut or copy clipboard operation.

Pascal

```
procedure CancelCutOrCopy;
```

Description

This method is used to stop any pending clipboard operation. No data is transfered nor are nodes deleted.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.176 TBaseVirtualTree.CancelEditNode Method

Cancel the current edit operation, if there is any.

Pascal

```
function CancelEditNode: Boolean;
```

Description

Used to stop the current edit operation.The node editor will get a CancelEdit call so that the node is not changed.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.177 TBaseVirtualTree.CanEdit Method

Determines whether a node can be edited or not.

Pascal

```
function CanEdit(Node: PVirtualNode; Column: TColumnIndex): Boolean; virtual;
```

Description

The method is called when the tree is about to start a node edit operation. Returns true if editing is allowed, otherwise false.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.178 TBaseVirtualTree.CanFocus Method

Support method to determine whether the tree window can receive the input focus.

Pascal

```
function CanFocus: Boolean;
```

Description

The method adds a check for the parent form of the control.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.179 TBaseVirtualTree.CanShowDragImage Method

Determines whether a drag image should be shown.

Pascal

```
function CanShowDragImage: Boolean; virtual;
```

Description

This overridable method is used to determine whether a drag image can be shown or not.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.180 TBaseVirtualTree.Change Method

Central method called when a node's selection state changes.

Pascal

```
procedure Change(Node: PVirtualNode); virtual;
```

Description

The Change method is called to trigger the change notifcation chain. Depending on the sync and the update states of the tree as well as the ChangeDelay(🏠 see TBaseVirtualTree.ChangeDelay Property, page 110) value either the application is directly notified about the change or a timer is started to accumulate several change events into one.

See Also

BeginSynch(🏠 see TBaseVirtualTree.BeginSynch Method , page 164), EndSynch(🏠 see TBaseVirtualTree.EndSynch Method , page 200), BeginUpdate(🏠 see TBaseVirtualTree.BeginUpdate Method , page 164), EndUpdate(🏠 see TBaseVirtualTree.EndUpdate Method , page 201), ChangeDelay(🏠 see TBaseVirtualTree.ChangeDelay Property, page 110)

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.181 TBaseVirtualTree.ChangeScale Method

Helper method called by the VCL when control resizing is due.

Pascal

```
procedure ChangeScale(M: Integer; D: Integer); override;
```

Description

ChangeScale is a method introduced by TControl. In Virtual Treeview it is responsible to change the tree's and the header's fonts as well as to compute the new default node height.

See Also

TVTHeader.ChangeScale(🏠 see TVTHeader.ChangeScale Method , page 551), DefaultNodeHeight(🏠 see TBaseVirtualTree.DefaultNodeHeight Property, page 113)

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.182 TBaseVirtualTree.CheckParentCheckState Method

Helper method for recursive check state changes.

Pascal

```
function CheckParentCheckState(Node: PVirtualNode; NewCheckState: TCheckState): Boolean; virtual;
```

Description

Checks all siblings of node to determine which check state Node's parent must get.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.183 TBaseVirtualTree.Clear Method

Clears the tree and removes all nodes.

Pascal

```
procedure Clear; virtual;
```

Description

All pending operations are stopped and the tree is ready to receive new nodes.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.184 TBaseVirtualTree.ClearChecked Method

Not documented.

Pascal

```
procedure ClearChecked;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.185 TBaseVirtualTree.ClearSelection Method

Removes all nodes from the current selection.

Pascal

```
procedure ClearSelection;
```

Description

ClearSelection empties the internal selection cache and resets the vsSelected state from all nodes, which were in this array.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.186 TBaseVirtualTree.ClearTempCache Method

Helper method to clear(see TBaseVirtualTree.Clear Method , page 168) the internal temporary node cache.

Pascal

```
procedure ClearTempCache; virtual;
```

Description

The internal node cache is used when more than one node is involved in certain operations (e.g. including a range of nodes into the current selection).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.187 TBaseVirtualTree.ColumnIsEmpty Method

Used to determine if a cell is considered as being empty.

Pascal

```
function ColumnIsEmpty(Node: PVirtualNode; Column: TColumnIndex): Boolean; virtual;
```

Description

An empty cell might be used for the automatic column spanning feature. Descentants can override this method to modify the tree's behavior.

See Also

toAutoSpanColumns

Class

TBaseVirtualTree Class(see page 88)

10.1.2.188 CopyTo

10.1.2.188.1 TBaseVirtualTree.CopyTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean)

Copies Source and all its child nodes to Target.

Pascal

```
function CopyTo(Source: PVirtualNode; Tree: TBaseVirtualTree; Mode: TVTNodeAttachMode; ChildrenOnly:
Boolean): PVirtualNode; overload;
function CopyTo(Source: PVirtualNode; Target: PVirtualNode; Mode: TVTNodeAttachMode; ChildrenOnly:
Boolean): PVirtualNode; overload;
```

Description

Mode is used to specify further where to add the new node actually (as sibling of Target or as child of Target). Result is the newly created node to which source has been copied if ChildrenOnly is False or just contains Target in the other case. ChildrenOnly determines whether to copy also the source node or only its child nodes.

The variant taking a tree reference as target can be used to transfer nodes to a different tree, without determining its root node first. However one can also pass in any virtual tree node as target, as long as it belongs to a tree. The owning tree is automatically determined.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.189 TBaseVirtualTree.CopyToClipBoard Method

Copies all currently selected nodes to the clipboard.

Pascal

```
procedure CopyToClipBoard; virtual;
```

Description

CopyToClipboard causes the tree to copy the currently selected nodes to the clipboard. Actually, Virtual Treeview maintains socalled delayed rendering. This means the participating nodes are marked as being in the current clipboard set (see vsCutOrCopy in TVirtualNodeStates(see TVirtualNodeStates Type, page 598)) and only an IDataObject interface is placed onto the clipboard but no data yet. This avoids not only possibly huge memory requirements but it also avoids rendering data in a format which is not necessary. The application which pastes the clipboard content later will get the IDataObject interface and requests the format it can handle. The actual data is then rendered when the target application calls IDataObject.GetData, which results in a call to RenderOLEData(see TBaseVirtualTree.RenderOLEData Method , page 231).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.190 TBaseVirtualTree.CountLevelDifference Method

Determines the level difference of two nodes.

Pascal

```
function CountLevelDifference(Node1: PVirtualNode; Node2: PVirtualNode): Integer; virtual;
```

Description

This method counts how many indentation levels the given nodes are apart. If both nodes have the same parent then the difference is 0 otherwise the result is basically GetNodeLevel(see TBaseVirtualTree.GetNodeLevel Method , page 210)(Node2) - GetNodeLevel(see TBaseVirtualTree.GetNodeLevel Method , page 210)(Node1), but with sign. If the result is negative then Node2 is less intended than Node1.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.191 TBaseVirtualTree.CountVisibleChildren Method

Determines the number of visible child nodes of the given node.

Pascal

```
function CountVisibleChildren(Node: PVirtualNode): Cardinal; virtual;
```

Description

CountVisibleChildren iterates through all child nodes of Node and counts how many of them have the vsVisible state set.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.192 TBaseVirtualTree.Create Constructor

Constructor of the control

Pascal

```
constructor Create(AOwner: TComponent); override;
```

Description

The constructor initializes certain properties to their default values.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.193 TBaseVirtualTree.CreateParams Method

Prepares the creation of the controls window handle.

Pascal

```
procedure CreateParams(var Params: TCreateParams); override;
```

Description

CreateParams is overriden to allow to set certain window styles for the control.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.194 TBaseVirtualTree.CreateWnd Method

Initializes data, which depends on the window handle.

Pascal

```
procedure CreateWnd; override;
```

Description

Some properties must be preset first after the window handle was created. CreateWnd is the perfect place for this.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.195 TBaseVirtualTree.CutToClipBoard Method

Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

Pascal

```
procedure CutToClipBoard; virtual;
```

Description

Similar to CopyToClipboard(see TBaseVirtualTree.CopyToClipBoard Method , page 169) only the nodes are deleted after they have been pasted into the target.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.196 TBaseVirtualTree.DefineProperties Method

Helper method to customize loading and saving persistent tree data.

Pascal

```
procedure DefineProperties(Filer: TFiler); override;
```

Description

There were heavy changes in some properties during development of VT. This method helps to make migration easier by reading old properties manually and put them into the new properties as appropriate. These old properties are never written again and silently disappear.

Another task of this method is to work around the problem that TCollection is not streamed correctly when using Visual Form Inheritance (VFI).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.197 TBaseVirtualTree.DeleteChildren Method

Removes all child nodes from the given node.

Pascal

```
procedure DeleteChildren(Node: PVirtualNode; ResetHasChildren: Boolean = False);
```

Description

The method works recursively: all grandchildren and their children are removed as well.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.198 TBaseVirtualTree.DeleteNode Method

Removes the given node from the tree.

Pascal

```
procedure DeleteNode(Node: PVirtualNode; Reindex: Boolean = True);
```

Description

This method deletes the given node. If the node was initialized or had gotten initial data via the AddChild(see TBaseVirtualTree.AddChild Method , page 160) or InsertNode(see TBaseVirtualTree.InsertNode Method , page 217) then the event OnFreeNode(see TBaseVirtualTree.OnFreeNode Event, page 139) is called to allow the application to free any user data attached to a node.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.199 TBaseVirtualTree.DeleteSelectedNodes Method

Removes all currently selected nodes form the tree.

Pascal

```
procedure DeleteSelectedNodes; virtual;
```

Description

All nodes in the current selection are affected.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.200 TBaseVirtualTree.Destroy Destructor

Destructor of the control.

Pascal

```
destructor Destroy; override;
```

Description

Frees any allocated data in the tree. All pending operations will be stopped and any remaining node is freed.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.201 TBaseVirtualTree.DetermineHiddenChildrenFlag Method

Determines whether all children of a given node are hidden.

Pascal

```
procedure DetermineHiddenChildrenFlag(Node: PVirtualNode); virtual;
```

Description

Virtual Treeview supports a feature, which is called node button auto hide. What happens is that when all children of a node are hidden then the expand button for this node is automatically removed. In order to know about the visibility state of the child nodes an internal flag is maintained, which allows to quickly decide about the button display. DetermineHidenChildren is the update method for cases where more than one child node changed.

See Also

vsVisible, toAutoHideButtons

Class

TBaseVirtualTree Class(see page 88)

10.1.2.202 TBaseVirtualTree.DetermineHiddenChildrenFlagAllNodes Method

Determines whether all children of all nodes are hidden.

Pascal

```
procedure DetermineHiddenChildrenFlagAllNodes; virtual;
```

Description

As extension to DeterminHiddenChildren this method iteratively determines the hidden children flag for all existing nodes in the tree. This is only used for large updates. No node will be initialized in this process.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.203 TBaseVirtualTree.DetermineHitPositionLTR Method

Determines the hit position within a node with left-to-right and right-to-left orientation.

Pascal

```
procedure DetermineHitPositionLTR(var HitInfo: THitInfo; Offset: Integer; Right: Integer; Alignment:
TAlignment); virtual;
procedure DetermineHitPositionRTL(var HitInfo: THitInfo; Offset: Integer; Right: Integer; Alignment:
TAlignment); virtual;
```

Description

This method, together with its counter part DetermineHitPositionRTL, is used in the process of figuring out where the a given position is located in relation to a node.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.204 TBaseVirtualTree.DetermineNextCheckState Method

Not documented.

Pascal

```
function DetermineNextCheckState(CheckType: TCheckType; CheckState: TCheckState): TCheckState; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.205 TBaseVirtualTree.DetermineScrollDirections Method

Not documented.

Pascal

```
function DetermineScrollDirections(X: Integer; Y: Integer): TScrollDirections; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.206 TBaseVirtualTree.DoAdvancedHeaderDraw Method

Not documented.

Pascal

```
procedure DoAdvancedHeaderDraw(var PaintInfo: THeaderPaintInfo; const Elements: THeaderPaintElements);
```

```
virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.207 TBaseVirtualTree.DoAfterCellPaint Method

Not documented.

Pascal

```
procedure DoAfterCellPaint(Canvas: TCanvas; Node: PVirtualNode; Column: TColumnIndex; CellRect:
TRect); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.208 TBaseVirtualTree.DoAfterItemErase Method

Not documented.

Pascal

```
procedure DoAfterItemErase(Canvas: TCanvas; Node: PVirtualNode; ItemRect: TRect); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.209 TBaseVirtualTree.DoAfterItemPaint Method

Not documented.

Pascal

```
procedure DoAfterItemPaint(Canvas: TCanvas; Node: PVirtualNode; ItemRect: TRect); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.210 TBaseVirtualTree.DoAfterPaint Method

Not documented.

Pascal

```
procedure DoAfterPaint(Canvas: TCanvas); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.211 TBaseVirtualTree.DoAutoScroll Method

Enables or disables the auto scroll timer.

Pascal

```
procedure DoAutoScroll(X: Integer; Y: Integer); virtual;
```

Description

This method determines whether the tree needs to be scrolled (the mouse is near the borders) and enables or disables the internal scroll timer which triggers the DoTimerScroll(see TBaseVirtualTree.DoTimerScroll Method , page 196) method.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.212 TBaseVirtualTree.DoBeforeCellPaint Method

Not documented.

Pascal

```
procedure DoBeforeCellPaint(Canvas: TCanvas; Node: PVirtualNode; Column: TColumnIndex; CellRect:
TRect); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.213 TBaseVirtualTree.DoBeforeDrag Method

Not documented.

Pascal

```
function DoBeforeDrag(Node: PVirtualNode; Column: TColumnIndex): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.214 TBaseVirtualTree.DoBeforeItemErase Method

Not documented.

Pascal

```
procedure DoBeforeItemErase(Canvas: TCanvas; Node: PVirtualNode; ItemRect: TRect; var Color: TColor;
var EraseAction: TItemEraseAction); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.215 TBaseVirtualTree.DoBeforeItemPaint Method

Not documented.

Pascal

```
function DoBeforeItemPaint(Canvas: TCanvas; Node: PVirtualNode; ItemRect: TRect): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.216 TBaseVirtualTree.DoBeforePaint Method

Not documented.

Pascal

```
procedure DoBeforePaint(Canvas: TCanvas); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.217 TBaseVirtualTree.DoCancelEdit Method

Called when the tree should stop editing without accepting changed values.

Pascal

```
function DoCancelEdit: Boolean; virtual;
```

Description

This method calls the edit link's IEditLink.CancelEdit method and stops the edit mode if this call returns True. If stopping is allowed then the event OnEditCancelled(see TBaseVirtualTree.OnEditCancelled Event, page 137) is triggered and a message is sent to release the edit link asynchronously.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.218 TBaseVirtualTree.DoCanEdit Method

Not documented.

Pascal

```
procedure DoCanEdit(Node: PVirtualNode; Column: TColumnIndex; var Allowed: Boolean); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.219 TBaseVirtualTree.DoChange Method

Not documented.

Pascal

```
procedure DoChange(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.220 TBaseVirtualTree.DoCheckClick Method

Not documented.

Pascal

```
procedure DoCheckClick(Node: PVirtualNode; NewCheckState: TCheckState); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.221 TBaseVirtualTree.DoChecked Method

Not documented.

Pascal

```
procedure DoChecked(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.222 TBaseVirtualTree.DoChecking Method

Not documented.

Pascal

```
function DoChecking(Node: PVirtualNode; var NewCheckState: TCheckState): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.223 TBaseVirtualTree.DoCollapsed Method

Not documented.

Pascal

```
procedure DoCollapsed(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.224 TBaseVirtualTree.DoCollapsing Method

Not documented.

Pascal

```
function DoCollapsing(Node: PVirtualNode): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.225 TBaseVirtualTree.DoColumnClick Method

Not documented.

Pascal

```
procedure DoColumnClick(Column: TColumnIndex; Shift: TShiftState); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.226 TBaseVirtualTree.DoColumnDblClick Method

Not documented.

Pascal

```
procedure DoColumnDblClick(Column: TColumnIndex; Shift: TShiftState); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.227 TBaseVirtualTree.DoColumnResize Method

Not documented.

Pascal

```
procedure DoColumnResize(Column: TColumnIndex); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.228 TBaseVirtualTree.DoCompare Method

Not documented.

Pascal

```
function DoCompare(Node1: PVirtualNode; Node2: PVirtualNode; Column: TColumnIndex): Integer; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.229 TBaseVirtualTree.DoCreateDataObject Method

Not documented.

Pascal

```
function DoCreateDataObject: IDataObject; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.230 TBaseVirtualTree.DoCreateDragManager Method

Not documented.

Pascal

```
function DoCreateDragManager: IVTDragManager; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.231 TBaseVirtualTree.DoCreateEditor Method

Not documented.

Pascal

```
function DoCreateEditor(Node: PVirtualNode; Column: TColumnIndex): IVTEditLink; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.232 TBaseVirtualTree.DoDragDrop Method

Not documented.

Pascal

```
procedure DoDragDrop(Source: TObject; DataObject: IDataObject; Formats: TFormatArray; Shift:
TShiftState; Pt: TPoint; var Effect: Integer; Mode: TDropMode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.233 TBaseVirtualTree.DoDragExpand Method

Not documented.

Pascal

```
procedure DoDragExpand; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.234 TBaseVirtualTree.DoDragging Method

Internal method which handles drag' drop.

Pascal

```
procedure DoDragging(P: TPoint); virtual;
```

Description

This method starts the OLE drag'n drop operation and returns after this operation is finished.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.235 TBaseVirtualTree.DoDragOver Method

Not documented.

Pascal

```
function DoDragOver(Source: TObject; Shift: TShiftState; State: TDragState; Pt: TPoint; Mode:
TDropMode; var Effect: Integer): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.236 TBaseVirtualTree.DoEdit Method

Initiates editing of the currently set focused column and edit node.

Pascal

```
procedure DoEdit; virtual;
```

Description

This method takes care for editor creation and initialization. You can look for tsEditing in TreeStates(see TBaseVirtualTree.TreeStates Property, page 158) to know whether editing is currently active.

See Also

tsEditing, OnCreateEditor(see TBaseVirtualTree.OnCreateEditor Event, page 134), IVTEditLink(see IVTEditLink Interface, page 707)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.237 TBaseVirtualTree.DoEndDrag Method

Not documented.

Pascal

```
procedure DoEndDrag(Target: TObject; X: Integer; Y: Integer); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.238 TBaseVirtualTree.DoEndEdit Method

Stops the current edit operation and takes over the new content.

Pascal

```
function DoEndEdit: Boolean; virtual;
```

Description

The method also sends a message to the tree window to asynchronously release the edit link which communicates to the actual editor. The edit link is responsible to propagate any changes made in its node editor to the tree.

Notes

TVirtualStringTree(see TVirtualStringTree Class, page 402) overrides this method to tell the application about the new caption by calling OnNewText.

See Also

DoEdit(see TBaseVirtualTree.DoEdit Method , page 182), OnNewText, EditNode(see TBaseVirtualTree.EditNode Method , page 200)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.239 TBaseVirtualTree.DoExpanded Method

Not documented.

Pascal

```
procedure DoExpanded(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.240 TBaseVirtualTree.DoExpanding Method

Not documented.

Pascal

```
function DoExpanding(Node: PVirtualNode): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.241 TBaseVirtualTree.DoFocusChange Method

Not documented.

Pascal

```
procedure DoFocusChange(Node: PVirtualNode; Column: TColumnIndex); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.242 TBaseVirtualTree.DoFocusChanging Method

Not documented.

Pascal

```
function DoFocusChanging(OldNode: PVirtualNode; NewNode: PVirtualNode; OldColumn: TColumnIndex;
NewColumn: TColumnIndex): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.243 TBaseVirtualTree.DoFocusNode Method

Internal method to set the focused node.

Pascal

```
procedure DoFocusNode(Node: PVirtualNode; Ask: Boolean); virtual;
```

Description

This methods is called by the property setter for the focused node as well as from other places to do the actual change. It takes the parameter Ask to optionally switch off (Ask = False) triggering the OnFocusChanging(see TBaseVirtualTree.OnFocusChanging Event, page 139) event.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.244 TBaseVirtualTree.DoFreeNode Method

Not documented.

Pascal

```
procedure DoFreeNode(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.245 TBaseVirtualTree.DoGetAnimationType Method

Determines the type of animation to be used.

Pascal

```
function DoGetAnimationType: THintAnimationType; virtual;
```

Description

Windows 98 and Windows 2000 introduced two ways of animating hints when they appear: a sliding window and a fading window. Virtual Treeview implements both animation types and also supports system dependent animations. This allows to use the animation type enabled in the particular system on which the tree currently runs. Additonally, there is a check for MMX to do a fallback if fade animation is specified but no MMX available. In this case sliding is used. Starting with Windows 2000 and Windows ME the hint animation can even be be switched off entirely. Also this case is handled by this method.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.246 TBaseVirtualTree.DoGetCursor Method

Not documented.

Pascal

```
procedure DoGetCursor(var Cursor: TCursor); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.247 TBaseVirtualTree.DoGetHeaderCursor Method

Not documented.

Pascal

```
procedure DoGetHeaderCursor(var Cursor: HCURSOR); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.248 TBaseVirtualTree.DoGetImageIndex Method

Not documented.

Pascal

```
function DoGetImageIndex(Node: PVirtualNode; Kind: TVTImageKind; Column: TColumnIndex; var Ghosted:
Boolean; var Index: Integer): TCustomImageList; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.249 TBaseVirtualTree.DoGetLineStyle Method

Not documented.

Pascal

```
procedure DoGetLineStyle(var Bits: Pointer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.250 TBaseVirtualTree.DoGetNodeHint Method

Not documented.

Pascal

```
function DoGetNodeHint(Node: PVirtualNode; Column: TColumnIndex; var LineBreakStyle:
TVTTooltipLineBreakStyle): WideString; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.251 TBaseVirtualTree.DoGetNodeTooltip Method

Not documented.

Pascal

```
function DoGetNodeTooltip(Node: PVirtualNode; Column: TColumnIndex; var LineBreakStyle:
TVTTooltipLineBreakStyle): WideString; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.252 TBaseVirtualTree.DoGetNodeWidth Method

Overridable method which always retuns 0.

Pascal

```
function DoGetNodeWidth(Node: PVirtualNode; Column: TColumnIndex; Canvas: TCanvas = nil): Integer;
virtual;
```

Description

Descentants override this method to return a value which describes the width of a node. This is the inner width of the node excluding tree lines etc. So TVirtualStringTree(see TVirtualStringTree Class, page 402) returns the width of the node caption (plus text margin).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.253 TBaseVirtualTree.DoGetPopupMenu Method

Overridable method which triggers the OnGetPopup event.

Pascal

```
function DoGetPopupMenu(Node: PVirtualNode; Column: TColumnIndex; Position: TPoint): TPopupMenu;
virtual;
```

Description

This method does an automatic parent traversal in the tree hierarchy to find a matching popup menu.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.254 TBaseVirtualTree.DoGetUserClipboardFormats Method

Not documented.

Pascal

```
procedure DoGetUserClipboardFormats(var Formats: TFormatEtcArray); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.255 TBaseVirtualTree.DoHeaderClick Method

Not documented.

Pascal

```
procedure DoHeaderClick(Column: TColumnIndex; Button: TMouseButton; Shift: TShiftState; X: Integer; Y:
Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.256 TBaseVirtualTree.DoHeaderDblClick Method

Not documented.

Pascal

```
procedure DoHeaderDblClick(Column: TColumnIndex; Button: TMouseButton; Shift: TShiftState; X: Integer;
Y: Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.257 TBaseVirtualTree.DoHeaderDragged Method

Not documented.

Pascal

```
procedure DoHeaderDragged(Column: TColumnIndex; OldPosition: TColumnPosition); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.258 TBaseVirtualTree.DoHeaderDraggedOut Method

Not documented.

Pascal

```
procedure DoHeaderDraggedOut(Column: TColumnIndex; DropPosition: TPoint); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.259 TBaseVirtualTree.DoHeaderDragging Method

Not documented.

Pascal

```
function DoHeaderDragging(Column: TColumnIndex): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.260 TBaseVirtualTree.DoHeaderDraw Method

Not documented.

Pascal

```
procedure DoHeaderDraw(Canvas: TCanvas; Column: TVirtualTreeColumn; R: TRect; Hover: Boolean; Pressed:
Boolean; DropMark: TVTDropMarkMode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.261 TBaseVirtualTree.DoHeaderDrawQueryElements Method

Not documented.

Pascal

```
procedure DoHeaderDrawQueryElements(var PaintInfo: THeaderPaintInfo; var Elements:
THeaderPaintElements); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.262 TBaseVirtualTree.DoHeaderMouseDown Method

Not documented.

Pascal

```
procedure DoHeaderMouseDown(Button: TMouseButton; Shift: TShiftState; X: Integer; Y: Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.263 TBaseVirtualTree.DoHeaderMouseMove Method

Not documented.

Pascal

```
procedure DoHeaderMouseMove(Shift: TShiftState; X: Integer; Y: Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.264 TBaseVirtualTree.DoHeaderMouseUp Method

Not documented.

Pascal

```
procedure DoHeaderMouseUp(Button: TMouseButton; Shift: TShiftState; X: Integer; Y: Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.265 TBaseVirtualTree.DoHotChange Method

Not documented.

Pascal

```
procedure DoHotChange(Old: PVirtualNode; New: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.266 TBaseVirtualTree.DoIncrementalSearch Method

Not documented.

Pascal

```
function DoIncrementalSearch(Node: PVirtualNode; const Text: WideString): Integer; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.267 TBaseVirtualTree.DoInitChildren Method

Not documented.

Pascal

```
procedure DoInitChildren(Node: PVirtualNode; var ChildCount: Cardinal); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.268 TBaseVirtualTree.DoInitNode Method

Not documented.

Pascal

```
procedure DoInitNode(Parent: PVirtualNode; Node: PVirtualNode; var InitStates:
TVirtualNodeInitStates); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.269 TBaseVirtualTree.DoKeyAction Method

Not documented.

Pascal

```
function DoKeyAction(var CharCode: Word; var Shift: TShiftState): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.270 TBaseVirtualTree.DoLoadUserData Method

Not documented.

Pascal

```
procedure DoLoadUserData(Node: PVirtualNode; Stream: TStream); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.271 TBaseVirtualTree.DoMeasureItem Method

Not documented.

Pascal

```
procedure DoMeasureItem(TargetCanvas: TCanvas; Node: PVirtualNode; var NodeHeight: Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.272 TBaseVirtualTree.DoNodeCopied Method

Not documented.

Pascal

```
procedure DoNodeCopied(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.273 TBaseVirtualTree.DoNodeCopying Method

Not documented.

Pascal

```
function DoNodeCopying(Node: PVirtualNode; NewParent: PVirtualNode): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.274 TBaseVirtualTree.DoNodeMoved Method

Not documented.

Pascal

```
procedure DoNodeMoved(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.275 TBaseVirtualTree.DoNodeMoving Method

Not documented.

Pascal

```
function DoNodeMoving(Node: PVirtualNode; NewParent: PVirtualNode): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.276 TBaseVirtualTree.DoPaintBackground Method

Not documented.

Pascal

```
function DoPaintBackground(Canvas: TCanvas; R: TRect): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.277 TBaseVirtualTree.DoPaintDropMark Method

Overridable method which draws the small line on top of a nodes image depending on the current drop state.

Pascal

```
procedure DoPaintDropMark(Canvas: TCanvas; Node: PVirtualNode; R: TRect); virtual;
```

Description

This method draws a simple polyline using Colors.DropMarkColor. Descentant can override this method to customize the appearance of the drop mark.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.278 TBaseVirtualTree.DoPaintNode Method

Overridable method which does nothing.

Pascal

```
procedure DoPaintNode(var PaintInfo: TVTPaintInfo); virtual;
```

Description

Descentants override this method to paint the content of the node. For instance string trees draw the node's caption.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.279 TBaseVirtualTree.DoPopupMenu Method

Overridable method which shows the popup menu for the given node.

Pascal

```
procedure DoPopupMenu(Node: PVirtualNode; Column: TColumnIndex; Position: TPoint); virtual;
```

Description

Node and Column describe the cell for which the menu should be shown. Position determines the place (in client coordinates of the tree window) where to show the menu.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.280 TBaseVirtualTree.DoRenderOLEData Method

Not documented.

Pascal

```
function DoRenderOLEData(const FormatEtcIn: TFormatEtc; out Medium: TStgMedium; ForClipboard:
Boolean): HRESULT; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.281 TBaseVirtualTree.DoReset Method

Not documented.

Pascal

```
procedure DoReset(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.282 TBaseVirtualTree.DoSaveUserData Method

Not documented.

Pascal

```
procedure DoSaveUserData(Node: PVirtualNode; Stream: TStream); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.283 TBaseVirtualTree.DoScroll Method

Overridable method which triggers the OnScroll(see TBaseVirtualTree.OnScroll Event, page 152) event.

Pascal

```
procedure DoScroll(DeltaX: Integer; DeltaY: Integer); virtual;
```

Description

This method is the ideal place if you want to synchronize other controls with the tree. The event is triggered whenever the tree is scrolled (by the user or programmatically). DeltaX and DeltaY contain the relative values the position changed about.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.284 TBaseVirtualTree.DoSetOffsetXY Method

Internal core routine to set the tree's scroll position.

Pascal

```
function DoSetOffsetXY(Value: TPoint; Options: TScrollUpdateOptions; ClipRect: PRect = nil): Boolean;
virtual;
```

Description

The method takes the Value structure which contains the new absolute scroll positions, both horizontal and vertical. Options specifies what should happen in the update process. A combination of the following values is possible:

- suoRepaintHeader, If suoUpdateNCArea is also set then invalidate the header to refresh its screen image, otherwise it is ignored.
- suoRepaintScrollbars, If suoUpdateNCArea is also set then repaint both scrollbars after updating them, otherwise it is ignored.
- suoScrollClientArea, Scroll and invalidate the proper part of the client area.
- suoUpdateNCArea, Update non-client area (scrollbars, header).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.285 TBaseVirtualTree.DoShowScrollbar Method

Not documented.

Pascal

```
procedure DoShowScrollbar(Bar: Integer; Show: Boolean); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.286 TBaseVirtualTree.DoStartDrag Method

Not documented.

Pascal

```
procedure DoStartDrag(var DragObject: TDragObject); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.287 TBaseVirtualTree.DoStateChange Method

Not documented.

Pascal

```
procedure DoStateChange(Enter: TVirtualTreeStates; Leave: TVirtualTreeStates = []); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.288 TBaseVirtualTree.DoStructureChange Method

Not documented.

Pascal

```
procedure DoStructureChange(Node: PVirtualNode; Reason: TChangeReason); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.289 TBaseVirtualTree.DoTimerScroll Method

Callback method which is triggered whenever the scroll timer fires.

Pascal

```
procedure DoTimerScroll; virtual;
```

Description

This method is called to do an automatic tree scroll when the user selects nodes with the mouse (multiselection only).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.290 TBaseVirtualTree.DoUpdating Method

Not documented.

Pascal

```
procedure DoUpdating(State: TVTUpdateState); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.291 TBaseVirtualTree.DoValidateCache Method

Not documented.

Pascal

```
function DoValidateCache: Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.292 TBaseVirtualTree.DragCanceled Method

Called by the VCL when a drag'n drop operation was canceled by the user.

Pascal

```
procedure DragCanceled; override;
```

Description

DragCanceled is used to do some housekeeping in the tree.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.293 TBaseVirtualTree.DragDrop Method

Helper method, which is used when a drag operation is finished.

Pascal

```
function DragDrop(const DataObject: IDataObject; KeyState: Integer; Pt: TPoint; var Effect: Integer):
HResult; virtual; reintroduce;
```

Description

This method is called by the TVTDragManager.Drop(see TVTDragManager.Drop Method , page 538) and prepares the list of available clipboard formats to be passed to DoDragDrop(see TBaseVirtualTree.DoDragDrop Method , page 181).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.294 TBaseVirtualTree.DragEnter Method

Not documented.

Pascal

```
function DragEnter(KeyState: Integer; Pt: TPoint; var Effect: Integer): HResult; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.295 TBaseVirtualTree.DragFinished Method

Called when a drag operation is finished (accepted or cancelled).

Pascal

```
procedure DragFinished; virtual;
```

Description

This method is nternally used ito make up for the swallowed mouse-up messages during drag' drop.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.296 TBaseVirtualTree.Dragging Method

Returns true if a drag'n drop operation is in progress.

Pascal

```
function Dragging: Boolean;
```

Description

The method returns true if currently a drag'n drop operation is in progress, which involves this tree view.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.297 TBaseVirtualTree.DragLeave Method

Not documented.

Pascal

```
procedure DragLeave; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.298 TBaseVirtualTree.DragOver Method

Not documented.

Pascal

```
function DragOver(Source: TObject; KeyState: Integer; DragState: TDragState; Pt: TPoint; var Effect:
Integer): HResult; virtual; reintroduce;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.299 TBaseVirtualTree.DrawDottedHLine Method

Not documented.

Pascal

```
procedure DrawDottedHLine(const PaintInfo: TVTPaintInfo; Left: Integer; Right: Integer; Top: Integer);
virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.300 TBaseVirtualTree.DrawDottedVLine Method

Not documented.

Pascal

```
procedure DrawDottedVLine(const PaintInfo: TVTPaintInfo; Top: Integer; Bottom: Integer; Left:
Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.301 TBaseVirtualTree.EditNode Method

Starts editing the given node if allowed to.

Pascal

```
function EditNode(Node: PVirtualNode; Column: TColumnIndex): Boolean; virtual;
```

Description

This method can be used by the application to manually start editiing of a particular node. Column determines hereby in which column the node should be edited. This parameter determines the target column regardless whether toExtendedFocus is set in TreeOptions.SelectionOptions or not. The given node must be enabled, otherwise edit start fails.

See Also

DoEdit(🏠 see TBaseVirtualTree.DoEdit Method , page 182)

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.302 TBaseVirtualTree.EndEditNode Method

Stops node editing if it was started before.

Pascal

```
function EndEditNode: Boolean;
```

Description

EndEditNode stops node editing and accepts the result (which must be set by the edit link).

See Also

Editors and editing(🏠 see page 41), EditNode(🏠 see TBaseVirtualTree.EditNode Method , page 200), DoEdit(🏠 see TBaseVirtualTree.DoEdit Method , page 182)

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.303 TBaseVirtualTree.EndSynch Method

Counterpart to BeginSynch(🏠 see TBaseVirtualTree.BeginSynch Method , page 164).

Pascal

```
procedure EndSynch;
```

Description

Counts down the internal synchronous mode counter and ends synchronous mode when this counter reaches zero.

See Also

BeginSynch(🏠 see TBaseVirtualTree.BeginSynch Method , page 164), BeginUpdate(🏠 see TBaseVirtualTree.BeginUpdate Method , page 164), EndUpdate(🏠 see TBaseVirtualTree.EndUpdate Method , page 201)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.304 TBaseVirtualTree.EndUpdate Method

Resets the update lock set by BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164).

Pascal

```
procedure EndUpdate;
```

Description

This method is the counterpart to BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164) and decreases the internal update count value. If this value reaches 0 then updates of the tree window will be allowed again. Additionally, some pending operations, which might be started during the update lock, are finished. This includes tasks like updating the selection list, validating the cache and sorting the tree if in auto sort mode.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.305 TBaseVirtualTree.ExecuteAction Method

Not documented.

Pascal

```
function ExecuteAction(Action: TBasicAction): Boolean; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.306 TBaseVirtualTree.FindNodeInSelection Method

Helper method to find the given node in the current selection.

Pascal

```
function FindNodeInSelection(P: PVirtualNode; var Index: Integer; LowBound: Integer; HighBound:
Integer): Boolean; virtual;
```

Description

This method does a binary search of the given node in the internal selection array which is sorted by memory references. The search is limited to the area given by LowBound and HighBound. If the node could be found then true is returned and Index is set to the found node position.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.307 TBaseVirtualTree.FinishChunkHeader Method

Not documented.

Pascal

```
procedure FinishChunkHeader(Stream: TStream; StartPos: Integer; EndPos: Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.308 TBaseVirtualTree.FinishCutOrCopy Method

Stops any pending cut or copy clipboard operation.

Pascal

```
procedure FinishCutOrCopy;
```

Description

This method is used by the tree (and can be used by the application too) to stop any pending cut or copy clipboard operation. If a cut operation is pending then nodes currently marked with the vsCutOrCopy state are deleted.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.309 TBaseVirtualTree.FlushClipboard Method

Renders all pending clipboard data.

Pascal

```
procedure FlushClipboard;
```

Description

Used to render the data which is currently on the clipboard and finishes so the delayed rendering. This method is useful if the tree is about to be destroyed but data from this tree is still on the clipboard and should stay there. If this method is not used then any pending clipboard operation is cancelled on tree destruction (by the tree instance which currently has data on the clipboard) and the clipboard itself is cleared.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.310 TBaseVirtualTree.FontChanged Method

Not documented.

Pascal

```
procedure FontChanged(AFont: TObject); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.311 TBaseVirtualTree.FullCollapse Method

Collapses all nodes in the tree.

Pascal

```
procedure FullCollapse(Node: PVirtualNode = nil); virtual;
```

Description

Call this method to bring all nodes in the tree into a collapsed state. This method is used to reset the vsExpanded state in all nodes in the tree. Nodes which are not yet initialized are also not expanded by definition and therefore do not need initialization.

See Also

FullExpand(see TBaseVirtualTree.FullExpand Method , page 203)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.312 TBaseVirtualTree.FullExpand Method

Expands all nodes in the tree.

Pascal

```
procedure FullExpand(Node: PVirtualNode = nil); virtual;
```

Description

Call this method to bring all nodes in the tree into an expanded state. This method expands every node in the tree and initializes nodes which are not yet initialized to expand them too if necessary. Since this will validate every node in the tree it is counterproductive and against the Virtual Paradigm(see The virtual paradigm, page 33).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.313 TBaseVirtualTree.GetBorderDimensions Method

Not documented.

Pascal

```
function GetBorderDimensions: TSize; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.314 TBaseVirtualTree.GetCheckImage Method

Not documented.

Pascal

```
function GetCheckImage(Node: PVirtualNode): Integer; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.315 TBaseVirtualTree.GetCheckImageListFor Method

Not documented.

Pascal

```
class function GetCheckImageListFor(Kind: TCheckImageKind): TCustomImageList; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.316 TBaseVirtualTree.GetColumnClass Method

Returns the class to be used to manage columns in the tree.

Pascal

```
function GetColumnClass: TVirtualTreeColumnClass; virtual;
```

Description

GetColumnClass is a special purpose method to return a certain class which is used by the tree for the columns. TVirtualBaseTree always returns TVirtualTreeColumn(see TVirtualTreeColumn Class, page 485) but descentants can override this method to return own classes.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.317 TBaseVirtualTree.GetControlsAlignment Method

Not documented.

Pascal

```
function GetControlsAlignment: TAlignment; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.318 TBaseVirtualTree.GetDisplayRect Method

Returns the visible region used by the given node in client coordinates.

Pascal

```
function GetDisplayRect(Node: PVirtualNode; Column: TColumnIndex; TextOnly: Boolean; Unclipped:
Boolean = False): TRect;
```

Description

If the given node cannot be found (because one of its parents is collapsed or it is invisible) then an empty rectangle is returned. If TextOnly is true then only the text bounds are returned, that is, the resulting rectangle's left and right border are updated according to the bidi mode, alignment and text width of the node. If Unclipped is true (which only makes sense if also TextOnly is true) then the calculated text rectangle is not clipped if the text does not entirely fit into the text space. This is special handling needed for hints.

If Column is NoColumn(see NoColumn Constant, page 686) then the entire client width is used before determining the node's width otherwise the bounds of the particular column are used.

Notes

Column must be a valid column and is used independent of whether the header is visible or not.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.319 TBaseVirtualTree.GetFirst Method

Group of node navigation functions.

Pascal

```
function GetFirst: PVirtualNode;
function GetFirstChild(Node: PVirtualNode): PVirtualNode;
function GetFirstCutCopy: PVirtualNode;
function GetFirstInitialized: PVirtualNode;
function GetFirstNoInit: PVirtualNode;
function GetFirstSelected: PVirtualNode;
function GetFirstVisible: PVirtualNode;
function GetFirstVisibleChild(Node: PVirtualNode): PVirtualNode;
function GetFirstVisibleChildNoInit(Node: PVirtualNode): PVirtualNode;
function GetFirstVisibleNoInit: PVirtualNode;
```

Description

This group of navigation functions is used to return the first node in the tree or first sub node with various properties.

GetFirst	First node in the tree with initialization.
GetFirstChild	First child node with initialization.
GetFirstCutCopy	First node in cut/copy set (no initialization needed).
GetFirstInitialized	First initialized node in the tree (no initialization needed).
GetFirstNoInit	First node in the tree without initialization.
GetFirstVisible	First visible node in the tree with initialization.
GetFirstVisibleChild	First visible child of a node with initialization.
GetFirstVisibleChildNoInit	First visible child of a node without initialization.

GetFirstVisibleNoInit	First visible node in the tree without initialization.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.320 TBaseVirtualTree.GetFirstChecked Method

Not documented.

Pascal

```
function GetFirstChecked(State: TCheckState): PVirtualNode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.321 TBaseVirtualTree.GetHeaderClass Method

Returns the header class to be used by the tree.

Pascal

```
function GetHeaderClass: TVTHeaderClass; virtual;
```

Description

As with several other classes in Virtual Treeview (e.g. drag manager, options etc.) also a customized header class is supported, which allows applications or descendant classes to implement their very own header class with special behavior. This is a further element to make Virtual Treeview as flexible as possible.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.322 TBaseVirtualTree.GetHintWindowClass Method

Not documented.

Pascal

```
function GetHintWindowClass: THintWindowClass; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.323 TBaseVirtualTree.GetHitTestInfoAt Method

Returns information about the node at the given position.

Pascal

```
procedure GetHitTestInfoAt(X: Integer; Y: Integer; Relative: Boolean; var HitInfo: THitInfo);
```

Description

This method returns information about the given hit position. If the position is not within the client area then the result is either of hiAbove, hiBelow, hiToLeft or hiToRight, depending on the side. If the position is within the client area but no node is hit (e.g. when the tree is empty) then hiNowhere is returned, otherwise the node is examined and HitInfo is filled with information about which node is hit by this position, which column is involved and where on the node is the hit (e.g. the caption, the expand/collapse button or the state image).

The parameter Relative is used to tell the method how to interpret the given coordinates. If this property is true then X and Y are given in client coordinates of the tree window, otherwise they represent absolute coordinates of the virtual tree image(see Tree image and tree window, page 38).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.324 TBaseVirtualTree.GetImageIndex Method

Not documented.

Pascal

```
procedure GetImageIndex(var Info: TVTPaintInfo; Kind: TVTImageKind; InfoIndex: TVTImageInfoIndex;
DefaultImages: TCustomImageList); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.325 TBaseVirtualTree.GetLast Method

Group of node navigation functions.

Pascal

```
function GetLast(Node: PVirtualNode = nil): PVirtualNode;
function GetLastInitialized(Node: PVirtualNode = nil): PVirtualNode;
function GetLastNoInit(Node: PVirtualNode = nil): PVirtualNode;
function GetLastChild(Node: PVirtualNode): PVirtualNode;
function GetLastChildNoInit(Node: PVirtualNode): PVirtualNode;
function GetLastVisible(Node: PVirtualNode = nil): PVirtualNode;
function GetLastVisibleChild(Node: PVirtualNode): PVirtualNode;
function GetLastVisibleChildNoInit(Node: PVirtualNode): PVirtualNode;
function GetLastVisibleNoInit(Node: PVirtualNode = nil): PVirtualNode;
```

Description

This group of navigation functions is used to return the last node in the tree or last sub node with various properties.

GetLast	Last node in the tree with initialization.
GetLastChild	Last child node with initialization.
GetLastChildNoInit	Last child node without initialiization.

GetLastInitialized	Last initialized node in the tree (no initialization needed).
GetLastNoInit	Last node in the tree without initialization.
GetLastVisible	Last visible node in the tree with initialization.
GetLastVisibleChild	Last visible child of a node with initialization.
GetLastVisibleChildNoInit	Last visible child of a node without initialization.
GetLastVisibleNoInit	Last visible node in the tree without initialization.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.326 TBaseVirtualTree.GetMaxColumnWidth Method

Returns the width of the largest node in the given column.

Pascal

```
function GetMaxColumnWidth(Column: TColumnIndex): Integer;
```

Description

This method is mainly used to determine a minimal width of the given column without having to shorten a node caption. Since the method has to go through all visible nodes and initialize them to learn about their width it might be time consuming to call this method and circumvents also the virtual approach of the tree.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.327 TBaseVirtualTree.GetMaxRightExtend Method

Determines the maximum with of the currently visible part of the tree.

Pascal

```
function GetMaxRightExtend: Cardinal; virtual;
```

Description

This method is similar to GetMaxColumnWidth(see TBaseVirtualTree.GetMaxColumnWidth Method , page 208), but determines the width of the tree if no columns are used. This method is used for determining the horizontal scroll range for the columnless case.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.328 TBaseVirtualTree.GetNativeClipboardFormats Method

Used to let descendants and the application add their own supported clipboard formats.

Pascal

```
procedure GetNativeClipboardFormats(var Formats: TFormatEtcArray); virtual;
```

Description

GetNativeClipboardFormats returns the supported clipboard formats of the tree in the native CF_* form as used in IDataObject. This includes all formats which are listed in the ClipboardFormats(see TBaseVirtualTree.ClipboardFormats

Property, page 112) property as well as any changes made by the OnGetUserClipboardFormats(see TBaseVirtualTree.OnGetUserClipboardFormats Event, page 142) event if a handler for it is attached.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.329 TBaseVirtualTree.GetNext Method

Group of node navigation functions.

Pascal

```
function GetNext(Node: PVirtualNode): PVirtualNode;
function GetNextCutCopy(Node: PVirtualNode): PVirtualNode;
function GetNextInitialized(Node: PVirtualNode): PVirtualNode;
function GetNextNoInit(Node: PVirtualNode): PVirtualNode;
function GetNextSelected(Node: PVirtualNode): PVirtualNode;
function GetNextSibling(Node: PVirtualNode): PVirtualNode;
function GetNextVisible(Node: PVirtualNode): PVirtualNode;
function GetNextVisibleNoInit(Node: PVirtualNode): PVirtualNode;
function GetNextVisibleSibling(Node: PVirtualNode): PVirtualNode;
function GetNextVisibleSiblingNoInit(Node: PVirtualNode): PVirtualNode;
```

Description

This group of navigation functions is used to return the next node relative to a given node in the tree with various properties.

GetNext	Next node in the tree with initialization.
GetNextCutCopy	Next node in the cut/copy set (no initialization needed).
GetNextInitialized	Next initialized node in the tree (no initialization needed).
GetNextNoInit	Next node in the tree without initialization.
GetNextSelected	Next selected node (no initialization needed).
GetNextSibling	Next sibling node with initialization.
GetNextVisible	Next visible node in the tree with initialization.
GetNextVisibleNoInit	Next visible node in the tree without initialization.
GetNextVisibleSibling	Next visible sibling node with initialization.
GetNextVisibleSiblingNoInit	Next visible sibling node without initialization.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.330 TBaseVirtualTree.GetNextChecked Method

Not documented.

Pascal

```
function GetNextChecked(Node: PVirtualNode; State: TCheckState = csCheckedNormal): PVirtualNode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.331 GetNodeAt

10.1.2.331.1 TBaseVirtualTree.GetNodeAt Method (Integer, Integer)

Not documented.

Pascal

```
function GetNodeAt(X: Integer; Y: Integer): PVirtualNode; overload;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.331.2 TBaseVirtualTree.GetNodeAt Method (Integer, Integer, Boolean, Integer)

Not documented.

Pascal

```
function GetNodeAt(X: Integer; Y: Integer; Relative: Boolean; var NodeTop: Integer): PVirtualNode;
overload;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.332 TBaseVirtualTree.GetNodeData Method

Returns the address of the user data area of the given node.

Pascal

```
function GetNodeData(Node: PVirtualNode): Pointer;
```

Description

GetNodeData returns the address of the user data area for Node. It is strongly recommended to use this method instead directly accessing @Node.Data. Some trees require internal data for their own use which is also stored after Node.Data and the actual user data (application data) follows then this internal data. GetNodeData takes care of this situation.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.333 TBaseVirtualTree.GetNodeLevel Method

Returns the indentation level of the given node.

Pascal

```
function GetNodeLevel(Node: PVirtualNode): Cardinal;
```

Description

GetNodeLevel returns the level of Node. This level is determined by the number of parent nodes (excluding the hidden root node). Top level nodes have the level 0, their direct child nodes have level 1 etc.

Class

TBaseVirtualTree Class(🏠 see page 88)

10.1.2.334 TBaseVirtualTree.GetOptionsClass Method

Customization helper to determine which options class the tree should use.

Pascal

```
function GetOptionsClass: TTreeOptionsClass; virtual;
```

Description

GetOptionsClass is a special purpose method to return a certain class which is used by the tree for its options. TVirtualBaseTree always returns TCustomVirtualTreeOptions(🏠 see TCustomVirtualTreeOptions Class, page 308) but descendants can override this method to return own classes.

For ease of use it makes much sense to always use the same name for the tree's options (which is TreeOptions(🏠 see TBaseVirtualTree.TreeOptions Property, page 158)). By using a customized options class, however, the wrong type is returned by this property. Hence it is meaningful to override TreeOptions(🏠 see TBaseVirtualTree.TreeOptions Property, page 158) and return the derived options class. To make this work the tree descendant must additionally provide new access methods for this property. An example can be seen in TVirtualStringTree(🏠 see TVirtualStringTree Class, page 402):

```
  TVirtualStringTree = class(TCustomVirtualStringTree)
  private
    function GetOptions: TStringTreeOptions;
    procedure SetOptions(const Value: TStringTreeOptions);
  protected
    function GetOptionsClass: TTreeOptionsClass; override;
  public
    property Canvas;
  published
    ...
    property TreeOptions: TStringTreeOptions read GetOptions write SetOptions;
    ...
  end;

  ...

//----------------- TVirtualStringTree
--------------------------------------------------------------------------------

function TVirtualStringTree.GetOptions: TStringTreeOptions;

begin
  Result := FOptions as TStringTreeOptions;
end;

//----------------------------------------------------------------------------------------------------
-----------------

procedure TVirtualStringTree.SetOptions(const Value: TStringTreeOptions);

begin
  FOptions.Assign(Value);
end;
```

```
//----------------------------------------------------------------------------------------------------
-----------------

function TVirtualStringTree.GetOptionsClass: TTreeOptionsClass;

begin
  Result := TStringTreeOptions;
end;
```

Class

TBaseVirtualTree Class(see page 88)

10.1.2.335 TBaseVirtualTree.GetPrevious Method

Group of node navigation functions.

Pascal

```
function GetPrevious(Node: PVirtualNode): PVirtualNode;
function GetPreviousInitialized(Node: PVirtualNode): PVirtualNode;
function GetPreviousNoInit(Node: PVirtualNode): PVirtualNode;
function GetPreviousSibling(Node: PVirtualNode): PVirtualNode;
function GetPreviousVisible(Node: PVirtualNode): PVirtualNode;
function GetPreviousVisibleNoInit(Node: PVirtualNode): PVirtualNode;
function GetPreviousVisibleSibling(Node: PVirtualNode): PVirtualNode;
function GetPreviousVisibleSiblingNoInit(Node: PVirtualNode): PVirtualNode;
```

Description

This group of navigation functions is used to return the previous node relative to a given node in the tree with various properties.

GetPrevious	Previous node in the tree with initialization.
GetPreviousInitialized	Previous initialized node in the tree (no initialization needed).
GetPreviousNoInit	Previous node in the tree without initialization.
GetPreviousSibling	Previous sibling node with initialization.
GetPreviousVisible	Previous visible node in the tree with initialization.
GetPreviousVisibleNoInit	Previous visible node in the tree without initialization.
GetPreviousVisibleSibling	Previous visible sibling node with initialization.
GetPreviousVisibleSiblingNoInit	Previous visible sibling node without initialization.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.336 TBaseVirtualTree.GetSortedCutCopySet Method

Returns a sorted list of nodes, which are marked for s cut or copy clipboard operation.

Pascal

```
function GetSortedCutCopySet(Resolve: Boolean): TNodeArray;
```

Description

Returns a list of nodes which are flagged with vsCutOrCopy, sorted in logical order, that is, as they appear in the tree. If Resolve is true then nodes which are children of other cut/copy nodes are not put into the new array. This feature is particularly important when doing drag'n drop as in this case all selected node plus their children need to be considered. A selected node, which is a child (grand child etc.) of another selected node is then automatically included and doesn't

need to be explicitly mentioned in the returned selection array.

Notes

The caller is responsible for freeing the array. Allocation is done here. Usually, though, freeing the array doesn't need additional attention as it is automatically freed by Delphi when it gets out of scope.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.337 TBaseVirtualTree.GetSortedSelection Method

Returns a sorted list of all currently selected nodes.

Pascal

```
function GetSortedSelection(Resolve: Boolean): TNodeArray;
```

Description

Returns a list of selected nodes sorted in logical order, that is, as they appear in the tree. If Resolve is true then nodes which are children of other selected nodes are not put into the new array. This feature is in particuar important when doing drag'n drop as in this case all selected node plus their children need to be considered. A selected node which is child (grand child etc.) of another selected node is then automatically included and doesn't need to be explicitely mentioned in the returned selection array.

Notes

The caller is responsible for freeing the array. Allocation is done here. Usually, though, freeing the array doesn't need additional attention as it is automatically freed by Delphi when it gets out of scope.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.338 TBaseVirtualTree.GetTextInfo Method

Helper method for node editors, hints etc.

Pascal

```
procedure GetTextInfo(Node: PVirtualNode; Column: TColumnIndex; const AFont: TFont; var R: TRect; var
Text: WideString); virtual;
```

Description

GetTextInfo is used to define a base access method for node data and the associated font from node editors and for hints.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.339 TBaseVirtualTree.GetTreeFromDataObject Method

OLE drag'n drop and clipboard support method.

Pascal

```
function GetTreeFromDataObject(const DataObject: IDataObject): TBaseVirtualTree; virtual;
```

Description

Returns the owner/sender of the given data object by means of a special clipboard format or nil if the sender is in another process or no virtual tree at all.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.340 TBaseVirtualTree.GetTreeRect Method

Returns the size of the virtual tree image.

Pascal

```
function GetTreeRect: TRect;
```

Description

GetTreeRect can be used to determine the full size of the tree image(see Tree image and tree window, page 38) as used for painting etc.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.341 TBaseVirtualTree.GetVisibleParent Method

Returns the first (nearest) parent node, which is visible.

Pascal

```
function GetVisibleParent(Node: PVirtualNode): PVirtualNode;
```

Description

GetVisibleParent returns the first (nearest) parent node of Node which is visible. This method is one of the seldom cases (if not the only one) where the hidden root node could be returned.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.342 TBaseVirtualTree.HandleHotTrack Method

Not documented.

Pascal

```
procedure HandleHotTrack(X: Integer; Y: Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.343 TBaseVirtualTree.HandleIncrementalSearch Method

Not documented.

Pascal

```
procedure HandleIncrementalSearch(CharCode: Word); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.344 TBaseVirtualTree.HandleMouseDblClick Method

Not documented.

Pascal

```
procedure HandleMouseDblClick(var Message: TWMMouse; const HitInfo: THitInfo); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.345 TBaseVirtualTree.HandleMouseDown Method

Not documented.

Pascal

```
procedure HandleMouseDown(var Message: TWMMouse; const HitInfo: THitInfo); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.346 TBaseVirtualTree.HandleMouseUp Method

Not documented.

Pascal

```
procedure HandleMouseUp(var Message: TWMMouse; const HitInfo: THitInfo); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.347 TBaseVirtualTree.HasAsParent Method

Determines if the given node has got another node as one of its parents.

Pascal

```
function HasAsParent(Node: PVirtualNode; PotentialParent: PVirtualNode): Boolean;
```

Description

Determines whether Node has got PotentialParent as one of its parents.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.348 TBaseVirtualTree.HasImage Method

Not documented.

Pascal

```
function HasImage(Node: PVirtualNode; Kind: TVTImageKind; Column: TColumnIndex): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.349 TBaseVirtualTree.HasPopupMenu Method

Determines whether there is a pop up menu assigned to the tree.

Pascal

```
function HasPopupMenu(Node: PVirtualNode; Column: TColumnIndex; Pos: TPoint): Boolean; virtual;
```

Description

This overridable method is used to determine whether there is a pop up menu assigned to the tree or can be retrieve via the OnGetPopupMenu(see TBaseVirtualTree.OnGetPopupMenu Event, page 142) event for a particular node. This is necessary for the tree to know how to deal with various condition in an mouse button down event.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.350 TBaseVirtualTree.InitChildren Method

Not documented.

Pascal

```
procedure InitChildren(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.351 TBaseVirtualTree.InitNode Method

Not documented.

Pascal

```
procedure InitNode(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.352 TBaseVirtualTree.InsertNode Method

Inserts a new node and returns it to the caller.

Pascal

```
function InsertNode(Node: PVirtualNode; Mode: TVTNodeAttachMode; UserData: Pointer = nil):
PVirtualNode;
```

Description

Adds a new node relative to Node. The final position is determined by Mode. UserData can be used to set the first 4 bytes of the user data area to an initial value, which can be used in OnInitNode(see TBaseVirtualTree.OnInitNode Event, page 147) and will also cause to trigger the OnFreeNode(see TBaseVirtualTree.OnFreeNode Event, page 139) event (if <> nil) even if the node is not yet "officially" initialized.

InsertNode is a compatibility method and will implicitly validates the given node if the new node is to be added as child node. This is however against the virtual paradigm and hence I dissuade from its usage.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.353 TBaseVirtualTree.InternalAddFromStream Method

Not documented.

Pascal

```
procedure InternalAddFromStream(Stream: TStream; Version: Integer; Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.354 InternalAddToSelection

10.1.2.354.1 TBaseVirtualTree.InternalAddToSelection Method (PVirtualNode, Boolean)

Not documented.

Pascal

```
function InternalAddToSelection(Node: PVirtualNode; ForceInsert: Boolean): Boolean; overload;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.354.2 TBaseVirtualTree.InternalAddToSelection Method (TNodeArray, Integer, Boolean)

Not documented.

Pascal

```
function InternalAddToSelection(const NewItems: TNodeArray; NewLength: Integer; ForceInsert: Boolean):
Boolean; overload;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.355 TBaseVirtualTree.InternalCacheNode Method

Not documented.

Pascal

```
procedure InternalCacheNode(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.356 TBaseVirtualTree.InternalClearSelection Method

Not documented.

Pascal

```
procedure InternalClearSelection; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.357 TBaseVirtualTree.InternalConnectNode Method

Not documented.

Pascal

```
procedure InternalConnectNode(Node: PVirtualNode; Destination: PVirtualNode; Target: TBaseVirtualTree;
Mode: TVTNodeAttachMode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.358 TBaseVirtualTree.InternalData Method

Returns the address of the internal data for a tree class.

Pascal

```
function InternalData(Node: PVirtualNode): Pointer;
```

Description

In TBaseVirtualTreeview this method returns nil but should be overridden in descendants to allow proper access to the internal data of Node if the descendant tree has allocated internal data.

See Also

Data handling(see page 39)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.359 TBaseVirtualTree.InternalDisconnectNode Method

Not documented.

Pascal

```
procedure InternalDisconnectNode(Node: PVirtualNode; KeepFocus: Boolean; Reindex: Boolean = True);
virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.360 TBaseVirtualTree.InternalRemoveFromSelection Method

Not documented.

Pascal

```
procedure InternalRemoveFromSelection(Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.361 TBaseVirtualTree.InvalidateCache Method

Empties the internal node cache and marks it as invalid.

Pascal

```
procedure InvalidateCache;
```

Description

Marks the internal node cache as being invalid. This will cause a cache validation run next time ValidateCache(see TBaseVirtualTree.ValidateCache Method , page 239) is called.

The internal node cache is used to speed up display in Virtual Treeview. It contains node references with a distance of CacheThreshold(see CacheThreshold Constant, page 673) nodes along with their vertical absolute position, which makes it possible to quickly find the position of a node for display, hit tests and so on.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.362 TBaseVirtualTree.InvalidateChildren Method

Invalidates all children of the given node.

Pascal

```
procedure InvalidateChildren(Node: PVirtualNode; Recursive: Boolean);
```

Description

Invalidates Node and its immediate children. If Recursive is true then all grandchildren are invalidated as well. The node itself is initialized if necessary and its child nodes are recreated (and initialized too if Recursive is true).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.363 TBaseVirtualTree.InvalidateColumn Method

Invalidates the client area part of a column.

Pascal

```
procedure InvalidateColumn(Column: TColumnIndex);
```

Description

Invalidates the client area part of a column.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.364 TBaseVirtualTree.InvalidateNode Method

Invalidates the given node.

Pascal

```
function InvalidateNode(Node: PVirtualNode): TRect; virtual;
```

Description

InvalidateNode initiates repaint of the given node by calling InvalidateRect with the node's display rectangel and returns this rectangle.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.365 TBaseVirtualTree.InvalidateToBottom Method

Invalidates the client area starting with the top position of the given node.

Pascal

```
procedure InvalidateToBottom(Node: PVirtualNode);
```

Description

InvalidateToBottom initiates repaint of client area starting at given node. If this node is not visible or not yet initialized then nothing happens.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.366 TBaseVirtualTree.InvertSelection Method

Inverts the current selection.

Pascal

```
procedure InvertSelection(VisibleOnly: Boolean);
```

Description

InvertSelection inverts the current selection, so nodes, which are selected become unselected and vice versa. If VisibleOnly is true then only visible nodes are considered.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.367 TBaseVirtualTree.IsEditing Method

Tells the caller whether the tree is currently in edit mode.

Pascal

```
function IsEditing: Boolean;
```

Description

Just a simple shortcut to test the tsEditing state.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.368 TBaseVirtualTree.IsMouseSelecting Method

Tell the caller whether the tree is currently in draw selection mode.

Pascal

```
function IsMouseSelecting: Boolean;
```

Description

IsMouseSelecting returns true if draw selection by the user is active or pending.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.369 TBaseVirtualTree.IterateSubtree Method

Iterator method to go through all nodes of a given sub tree.

Pascal

```
function IterateSubtree(Node: PVirtualNode; Callback: TVTGetNodeProc; Data: Pointer; Filter:
TVirtualNodeStates = []; DoInit: Boolean = False; ChildNodesOnly: Boolean = False): PVirtualNode;
```

Description

IterateSubtree iterates through all children and grandchildren etc. of Node (or the entire tree if Node = nil) and calls for each node the provided callback method (which must not be empty). Filter determines which nodes are to be considered (an empty set denotes all nodes). If DoInit is true then nodes which aren't initialized yet will be initialized.

During execution of the callback the application can set Abort to true. In this case the iteration is stopped and the last accessed node (the one on which the callback set Abort to true) is returned to the caller. Otherwise (no abort) nil is returned.

Notes

An application should not modify the content of the tree (e.g. delete nodes) during the iteration, otherwise the outcome is unpredictable and may result in an access violation.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.370 TBaseVirtualTree.Loaded Method

Not documented.

Pascal

```
procedure Loaded; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.371 TBaseVirtualTree.LoadFromFile Method

Loads previously streamed out tree data back in again.

Pascal

```
procedure LoadFromFile(const FileName: TFileName); virtual;
procedure LoadFromStream(Stream: TStream); virtual;
```

Description

LoadFromFile clears the current content of the tree and loads a new structure from the given file.

See Also

AddFromStream(see TBaseVirtualTree.AddFromStream Method , page 161)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.372 TBaseVirtualTree.MainColumnChanged Method

Not documented.

Pascal

```
procedure MainColumnChanged; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.373 TBaseVirtualTree.MarkCutCopyNodes Method

Not documented.

Pascal

```
procedure MarkCutCopyNodes; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.374 TBaseVirtualTree.MeasureItemHeight Method

Not documented.

Pascal

```
procedure MeasureItemHeight(const Canvas: TCanvas; Node: PVirtualNode);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.375 TBaseVirtualTree.MouseMove Method

Not documented.

Pascal

```
procedure MouseMove(Shift: TShiftState; X: Integer; Y: Integer); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.376 MoveTo

10.1.2.376.1 TBaseVirtualTree.MoveTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean)

Moves Source and all its child nodes to Target.

Pascal

```
procedure MoveTo(Source: PVirtualNode; Target: PVirtualNode; Mode: TVTNodeAttachMode; ChildrenOnly:
Boolean); overload;
procedure MoveTo(Node: PVirtualNode; Tree: TBaseVirtualTree; Mode: TVTNodeAttachMode; ChildrenOnly:
Boolean); overload;
```

Description

Moves the given node (and all its children) to Target. Source must belong to the tree instance which calls this MoveTo method. Mode determines how to connect Source to Target. This method might involve a change of the tree if Target belongs to a different tree than Source.

The variant taking a tree reference as target can be used to transfer nodes to a different tree, without determining its root node first. However one can also pass in any virtual tree node as target, as long as it belongs to a tree. The owning tree is automatically determined and an optimized path is taken if the operation happens within one tree. In this case

simply the source node is disconnected from the old place and reconnected at the new location.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.377 TBaseVirtualTree.Notification Method

Not documented.

Pascal

```
procedure Notification(AComponent: TComponent; Operation: TOperation); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.378 TBaseVirtualTree.OriginalWMNCPaint Method

Not documented.

Pascal

```
procedure OriginalWMNCPaint(DC: HDC); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.379 TBaseVirtualTree.Paint Method

TControl's Paint method used here to display the tree.

Pascal

```
procedure Paint; override;
```

Description

Overriden method to paint the tree image. The actual work is however done in PaintTree(see TBaseVirtualTree.PaintTree Method , page 226).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.380 TBaseVirtualTree.PaintCheckImage Method

Not documented.

Pascal

```
procedure PaintCheckImage(const PaintInfo: TVTPaintInfo); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.381 TBaseVirtualTree.PaintImage Method

Not documented.

Pascal

```
procedure PaintImage(var PaintInfo: TVTPaintInfo; ImageInfoIndex: TVTImageInfoIndex; DoOverlay:
Boolean); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.382 TBaseVirtualTree.PaintNodeButton Method

Not documented.

Pascal

```
procedure PaintNodeButton(Canvas: TCanvas; Node: PVirtualNode; const R: TRect; ButtonX: Integer;
ButtonY: Integer; BidiMode: TBiDiMode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.383 TBaseVirtualTree.PaintSelectionRectangle Method

Not documented.

Pascal

```
procedure PaintSelectionRectangle(Target: TCanvas; WindowOrgX: Integer; const SelectionRect: TRect;
TargetRect: TRect); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.384 TBaseVirtualTree.PaintTree Method

Main paint routine for the tree image.

Pascal

```
procedure PaintTree(TargetCanvas: TCanvas; Window: TRect; Target: TPoint; PaintOptions:
TVTInternalPaintOptions; PixelFormat: TPixelFormat = pfDevice);
```

Description

PaintTree is the core paint routine used to draw any part of the tree image to any canvas. It is responsible for maintaining the paint cycles per node as well as coordinating drawing of the various parts of the tree image. TargetCanvas is the canvas to which to draw the tree image. This is usually the tree window itself but could well be a bitmap or printer canvas. Window determines which part of the entire tree image to draw. The full size of the virtual image is determined by GetTreeRect(see TBaseVirtualTree.GetTreeRect Method , page 214). Target is the position in TargetCanvas where to draw the tree part specified by Window. PaintOptions determines what of the tree to draw. For different tasks usually different parts need to be drawn, with a full image in the window, selected only nodes for a drag image etc.

See Also

Tree image and tree window(see page 38)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.385 TBaseVirtualTree.PaintTreeLines Method

Not documented.

Pascal

```
procedure PaintTreeLines(const PaintInfo: TVTPaintInfo; VAlignment: Integer; IndentSize: Integer;
LineImage: TLineImage); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.386 TBaseVirtualTree.PanningWindowProc Method

Not documented.

Pascal

```
procedure PanningWindowProc(var Message: TMessage); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.387 TBaseVirtualTree.PasteFromClipboard Method

Inserts the content of the clipboard into the tree.

Pascal

```
function PasteFromClipboard: Boolean; virtual;
```

Description

PasteFromClipboar reads what is currently on the clipboard into the tree (if the format is supported). If the application wants to have text or special formats to be inserted then it must implement its own code (OLE). Here only the native tree format is accepted.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.388 TBaseVirtualTree.PrepareDragImage Method

Not documented.

Pascal

```
procedure PrepareDragImage(HotSpot: TPoint; const DataObject: IDataObject);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.389 TBaseVirtualTree.Print Method

Not documented.

Pascal

```
procedure Print(Printer: TPrinter; PrintHeader: Boolean);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.390 TBaseVirtualTree.ProcessDrop Method

Helper method to ease OLE drag'n drop operations.

Pascal

```
function ProcessDrop(DataObject: IDataObject; TargetNode: PVirtualNode; var Effect: Integer; Mode:
TVTNodeAttachMode): Boolean;
```

Description

ProcessDrop can be used in a OnDragDrop(see TBaseVirtualTree.OnDragDrop Event, page 135) handler to let the tree view handle a drop operation of native tree data. The method only prepares some variables and calls then the more universal ProcessOLEData(see TBaseVirtualTree.ProcessOLEData Method , page 229) method.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.391 TBaseVirtualTree.ProcessOLEData Method

Takes serialized OLE tree data and reconstructs the former structure.

Pascal

```
function ProcessOLEData(Source: TBaseVirtualTree; DataObject: IDataObject; TargetNode: PVirtualNode;
Mode: TVTNodeAttachMode; Optimized: Boolean): Boolean;
```

Description

PrcessOLEData recreates the (sub) tree structure serialized into memory and provided by DataObject. The new nodes are attached to the passed node or the hidden root node if TargetNode is nil, according to Mode. Optimized can be set to true if the entire operation happens within the same process (i.e. sender and receiver of the OLE operation are located in the same process). Optimized = true makes only sense if the operation to carry out is a move hence it is also the indication of the operation to be done here. Source is the source of the OLE data and only of use (and usually assigned) when an OLE operation takes place in the same application.

The function returns true on success, i.e. the CF_VIRTUALTREE(see CF_VIRTUALTREE Variable, page 658) format is supported by the data object and the structure could be recreated, otherwise false.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.392 TBaseVirtualTree.ReadChunk Method

Not documented.

Pascal

```
function ReadChunk(Stream: TStream; Version: Integer; Node: PVirtualNode; ChunkType: Integer;
ChunkSize: Integer): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.393 TBaseVirtualTree.ReadNode Method

Not documented.

Pascal

```
procedure ReadNode(Stream: TStream; Version: Integer; Node: PVirtualNode); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.394 TBaseVirtualTree.RedirectFontChangeEvent Method

Not documented.

Pascal

```
procedure RedirectFontChangeEvent(Canvas: TCanvas); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.395 TBaseVirtualTree.ReinitChildren Method

Forces all child nodes of Node to be reinitialized.

Pascal

```
procedure ReinitChildren(Node: PVirtualNode; Recursive: Boolean); virtual;
```

Description

ReinitChildren forces all child nodes of Node to be reinitialized. If Recursive is true then also the grandchildren are reinitialized.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.396 TBaseVirtualTree.ReinitNode Method

Forces a reinitialization of the given node.

Pascal

```
procedure ReinitNode(Node: PVirtualNode; Recursive: Boolean); virtual;
```

Description

ReinitNode forces Node and all its children (if Recursive is true) to be initialized again without modifying any data in the nodes nor deleting children (unless the application requests a different amount).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.397 TBaseVirtualTree.RemoveFromSelection Method

Removes the given node from the current selection.

Pascal

```
procedure RemoveFromSelection(Node: PVirtualNode); virtual;
```

Description

Removes the vsSelected style from Node's states and also removes Node from the internal selection array.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.398 TBaseVirtualTree.RenderOLEData Method

Renders pending OLE data.

Pascal

```
function RenderOLEData(const FormatEtcIn: TFormatEtc; out Medium: TStgMedium; ForClipboard: Boolean):
HResult; virtual;
```

Description

RenderOLData is called by TVTDataObject.GetData(see TVTDataObject.GetData Method , page 527) when a consumer of clipboard data actually requests the data. The base tree view only renders the native tree format, which is a chunk based stream of node data. The format to be rendered is specified in FormatEtcIn.cfFormat and is one of the formats which are returned from GetNativeClipboardFormats(see TBaseVirtualTree.GetNativeClipboardFormats Method , page 208).

Descendants may override RenderOLEData in order to render other formats like HTML text. In TBaseVirtualTreeview this method calls the OnRenderOLEData(see TBaseVirtualTree.OnRenderOLEData Event, page 150) event for all formats, except CF_VIRTUALTREE(see CF_VIRTUALTREE Variable, page 658).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.399 TBaseVirtualTree.RepaintNode Method

Causes the treeview to repaint the given node.

Pascal

```
procedure RepaintNode(Node: PVirtualNode);
```

Description

RepaintNode causes an immediate repaint of Node and returns once repainting has finished.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.400 TBaseVirtualTree.ResetNode Method

Resets the given node to uninitialized.

Pascal

```
procedure ResetNode(Node: PVirtualNode); virtual;
```

Description

ResetNode deletes all children of Node and marks it as being uninitialized.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.401 TBaseVirtualTree.ResetRangeAnchor Method

Not documented.

Pascal

```
procedure ResetRangeAnchor; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.402 TBaseVirtualTree.RestoreFontChangeEvent Method

Not documented.

Pascal

```
procedure RestoreFontChangeEvent(Canvas: TCanvas); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.403 TBaseVirtualTree.SaveToFile Method

Saves the entire content of the tree into a file or stream.

Pascal

```
procedure SaveToFile(const FileName: TFileName);
procedure SaveToStream(Stream: TStream; Node: PVirtualNode = nil); virtual;
```

Description

Saves the entire content of the tree into a file or stream.

See Also

LoadFromStream(see TBaseVirtualTree.LoadFromFile Method , page 223), AddFromStream(see TBaseVirtualTree.AddFromStream Method , page 161)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.404 TBaseVirtualTree.ScrollIntoView Method

Scrolls the tree so that the given node comes in the client area.

Pascal

```
function ScrollIntoView(Node: PVirtualNode; Center: Boolean; Horizontally: Boolean = False): Boolean;
```

Description

ScrollIntoView scrolls the tree so that the given node is in the client area and returns true if the tree really has been scrolled (e.g. to avoid further updates) else it returns false. If extened focus is enabled then the tree will also horizontally scrolled if needed. All collapsed parents of the node are expanded, forming so a visible path to Node.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.405 TBaseVirtualTree.SelectAll Method

Selects all nodes in the tree.

Pascal

```
procedure SelectAll(VisibleOnly: Boolean);
```

Description

SelectAll select all existing nodes in the tree. If VisibleOnly is true then only visible nodes are selected.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.406 TBaseVirtualTree.SelectNodes Method

Selects a range of nodes.

Pascal

```
procedure SelectNodes(StartNode: PVirtualNode; EndNode: PVirtualNode; AddOnly: Boolean); virtual;
```

Description

SelectNodes selects a range of nodes and unselects all other possibly selected nodes which are not in this range if AddOnly is false. EndNode must be visible while StartNode does not necessarily, as in the case where the last focused node is the start node but it is a child of a node which has been collapsed previously. In this case the first visible parent node is used as start node. StartNode can be nil in which case the very first node in the tree is used.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.407 TBaseVirtualTree.SetBiDiMode Method

Not documented.

Pascal

```
procedure SetBiDiMode(Value: TBiDiMode); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.408 TBaseVirtualTree.SetFocusedNodeAndColumn Method

Not documented.

Pascal

```
procedure SetFocusedNodeAndColumn(Node: PVirtualNode; Column: TColumnIndex); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.409 TBaseVirtualTree.SkipNode Method

Not documented.

Pascal

```
procedure SkipNode(Stream: TStream); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.410 TBaseVirtualTree.Sort Method

Sorts the given node.

Pascal

```
procedure Sort(Node: PVirtualNode; Column: TColumnIndex; Direction: TSortDirection; DoInit: Boolean =
True); virtual;
```

Description

Sort sorts the child nodes of Node. The application is queried about how to sort via the OnCompareNodes(see TBaseVirtualTree.OnCompareNodes Event, page 133) event. Column is simply passed to the the compare function so the application can also sort in a particular column. In order to free the application from taking care about the sort direction the parameter Direction is used. This way the application can always compare as would the node be sorted in increasing direction , while Sort reorders nodes according to this flag.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.411 TBaseVirtualTree.SortTree Method

Sorts the entire tree view.

Pascal

```
procedure SortTree(Column: TColumnIndex; Direction: TSortDirection; DoInit: Boolean = True);
```

Description

SortTree sorts the entire tree by applying Sort to every node which has got children.

Notes

This method initializes all nodes in the tree which may not only take quite a while but is also against the virtual paradigm(see The virtual paradigm, page 33)

and therefore usually not recommended.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.412 TBaseVirtualTree.StartWheelPanning Method

Not documented.

Pascal

```
procedure StartWheelPanning(Position: TPoint); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.413 TBaseVirtualTree.StopWheelPanning Method

Not documented.

Pascal

```
procedure StopWheelPanning; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.414 TBaseVirtualTree.StructureChange Method

Not documented.

Pascal

```
procedure StructureChange(Node: PVirtualNode; Reason: TChangeReason); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.415 TBaseVirtualTree.SuggestDropEffect Method

Not documented.

Pascal

```
function SuggestDropEffect(Source: TObject; Shift: TShiftState; Pt: TPoint; AllowedEffects: Integer):
Integer; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.416 TBaseVirtualTree.ToggleNode Method

Changes a node's expand state to the opposite state.

Pascal

```
procedure ToggleNode(Node: PVirtualNode);
```

Description

Toggle node expands Node if it is collapsed currently and vice versa.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.417 TBaseVirtualTree.ToggleSelection Method

Toggles the selection state of a range of nodes.

Pascal

```
procedure ToggleSelection(StartNode: PVirtualNode; EndNode: PVirtualNode); virtual;
```

Description

ToggleSelection switchs the selection state of a range of nodes, so selected nodes become unselected and vice versa. This method is specifically designed to help selecting ranges with the keyboard and considers therefore the range anchor.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.418 TBaseVirtualTree.UnselectNodes Method

Deselects a range of nodes.

Pascal

```
procedure UnselectNodes(StartNode: PVirtualNode; EndNode: PVirtualNode); virtual;
```

Description

UnselectNodes deselects a given range of nodes. EndNode must be visible while StartNode is not required to be so as in the case where the last focused node is the start node but it is a child of a node which has been collapsed previously. In this case the first visible parent node is used as start node. StartNode can be nil in which case the very first node in the

tree is used.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.419 TBaseVirtualTree.UpdateAction Method

Not documented.

Pascal

```
function UpdateAction(Action: TBasicAction): Boolean; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.420 TBaseVirtualTree.UpdateDesigner Method

Not documented.

Pascal

```
procedure UpdateDesigner; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.421 TBaseVirtualTree.UpdateEditBounds Method

Not documented.

Pascal

```
procedure UpdateEditBounds; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.422 TBaseVirtualTree.UpdateHeaderRect Method

Not documented.

Pascal

```
procedure UpdateHeaderRect; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.423 TBaseVirtualTree.UpdateScrollBars Method

Applies changes to the horizontal and vertical scrollbars.

Pascal

```
procedure UpdateHorizontalScrollBar(DoRepaint: Boolean);
procedure UpdateScrollBars(DoRepaint: Boolean); virtual;
procedure UpdateVerticalScrollBar(DoRepaint: Boolean);
```

Description

UpdateScrollbars (and its counterparts for vertical and horizontal scrollbars) is the core method to set the scrollbar's properties like range, page size etc.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.424 TBaseVirtualTree.UpdateWindowAndDragImage Method

Not documented.

Pascal

```
procedure UpdateWindowAndDragImage(const Tree: TBaseVirtualTree; TreeRect: TRect; UpdateNCArea:
Boolean; ReshowDragImage: Boolean); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.425 TBaseVirtualTree.UseRightToLeftReading Method

Helper method for right-to-left layout.

Pascal

```
function UseRightToLeftReading: Boolean;
```

Description

UseRightToLeftReading had to be overriden in order to overcome a limitation introduced by the VCL. The VCL only allows a window to be in right-to-left reading order if the operating system is prepared to handle this (e.g. an arabic Windows 98). Virtual Treeview however does most of the RTL stuff handle itself, also on non-RTL system.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.426 TBaseVirtualTree.ValidateCache Method

Initiates the validation of the internal node cache.

Pascal

```
procedure ValidateCache; virtual;
```

Description

If the node cache is marked as being invalid then this method puts the tree into the worker thread's list and awakes then the thread so that the validation is performed in the background.

See Also

InvalidateCache(see TBaseVirtualTree.InvalidateCache Method , page 220)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.427 TBaseVirtualTree.ValidateChildren Method

Validates all children of a given node.

Pascal

```
procedure ValidateChildren(Node: PVirtualNode; Recursive: Boolean);
```

Description

ValidateChildren ensures that the children of the given node (and all their children, if Recursive is true) are initialized. Node must already be initialized. If nil is passed to the method the hidden root node is used (which makes only sense if Recursive is true, in which case the entire tree is validated).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.428 TBaseVirtualTree.ValidateNode Method

Validates a given node.

Pascal

```
procedure ValidateNode(Node: PVirtualNode; Recursive: Boolean);
```

Description

ValidateNode ensures that the given node (and all its children, if Recursive is true) are initialized. If Node is nil then the hidden root node is used (which makes only sense if Recursive is true, in which case the entire tree is validated).

Class

TBaseVirtualTree Class(see page 88)

10.1.2.429 TBaseVirtualTree.ValidateNodeDataSize Method

Helper method for node data size initalization.

Pascal

```
procedure ValidateNodeDataSize(var Size: Integer); virtual;
```

Description

ValidateNodeDataSize is called from MakeNewNode if the currently set node data size is -1, which indicates it has not yet been determined. The method calls the event OnGetNodeDataSize(see TBaseVirtualTree.OnGetNodeDataSize Event, page 142) allowing so the application to compute now its data requirement.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.430 TBaseVirtualTree.WndProc Method

Redirected window procedure to do some special processing.

Pascal

```
procedure WndProc(var Message: TMessage); override;
```

Description

WndProc has been overriden to allow the header to handle certain messages (which are forwarded by the tree) as well as to do some other special handling internal to the tree.

Class

TBaseVirtualTree Class(see page 88)

10.1.2.431 TBaseVirtualTree.WriteChunks Method

Writes the core chunks for the given node to the given stream.

Pascal

```
procedure WriteChunks(Stream: TStream; Node: PVirtualNode); virtual;
```

Description

WriteChunks is part of the streaming system in Virtual Treeview and writes the core chunks for Node into Stream. Descentants can optionally override this method to add other node specific chunks. This streaming is used when the tree must be saved to disk or a stream used e.g. for clipboard operations.

Notes

Keep in mind that this method is also called for the hidden root node. Using this fact in descendants you can create(see TBaseVirtualTree.Create Constructor , page 170) a

kind of "global" chunk set not directly bound to a specific node.

See Also

WriteNode(see TBaseVirtualTree.WriteNode Method , page 241), SaveToStream(see TBaseVirtualTree.SaveToFile Method , page 232)

Class

TBaseVirtualTree Class(see page 88)

10.1.2.432 TBaseVirtualTree.WriteNode Method

Writes the cover (envelop) chunk for the given node to the given stream.

Pascal

```
procedure WriteNode(Stream: TStream; Node: PVirtualNode); virtual;
```

Description

WriteNode writes the cover chunk for Node to Stream and initiates writing child nodes and chunks. This method is part of the streaming system used in Virtual Treeview.

See Also

WriteChunks(see TBaseVirtualTree.WriteChunks Method , page 240), WriteToStream

Class

TBaseVirtualTree Class(see page 88)

10.1.3 TBufferedString Class

```
TBufferedString = class;
```

Group

Classes(see page 86)

Methods

Add(see TBufferedString.Add Method , page 242)
Not documented.

AddNewLine(see TBufferedString.AddNewLine Method , page 242)
Not documented.

Destroy(see TBufferedString.Destroy Destructor , page 242)
Not documented.

Properties

AsString(see TBufferedString.AsString Property, page 242)
Not documented.

Legend

public

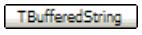
Property

read only

Method

virtual

Class Hierarchy



File

VirtualTrees

10.1.3.1 TBufferedString.AsString Property

Not documented.

Pascal

```
property AsString: string;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBufferedString Class(see page 241)

10.1.3.2 TBufferedString.Add Method

Not documented.

Pascal

```
procedure Add(const S: string);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBufferedString Class(see page 241)

10.1.3.3 TBufferedString.AddNewLine Method

Not documented.

Pascal

```
procedure AddNewLine;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBufferedString Class(see page 241)

10.1.3.4 TBufferedString.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBufferedString Class(see page 241)

10.1.4 TClipboardFormatList Class

Not documented.

Pascal

```
TClipboardFormatList = class;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes(see page 86)

Methods

Add(see TClipboardFormatList.Add Method , page 243)
Adds the given data to the internal list.

Clear(see TClipboardFormatList.Clear Method , page 244)
Not documented.

Create(see TClipboardFormatList.Create Constructor , page 244)
Not documented.

Destroy(see TClipboardFormatList.Destroy Destructor , page 244)
Not documented.

EnumerateFormats(see TClipboardFormatList.EnumerateFormats Method (TVirtualTreeClass, TFormatEtcArray, TClipboardFormats), page 245)
Returns a list of format records for the given class.

FindFormat(see TClipboardFormatList.FindFormat Method (Word, string), page 245)
Not documented.

Legend

public

Method

virtual

Class Hierarchy

File

VirtualTrees

10.1.4.1 TClipboardFormatList.Add Method

Adds the given data to the internal list.

Pascal

```
procedure Add(FormatString: string; AClass: TVirtualTreeClass; Priority: Cardinal; AFormatEtc:
TFormatEtc);
```

Description

The priority value is used to sort formats for importance. Larger priority values mean less priority.

Class

TClipboardFormatList Class(see page 243)

10.1.4.2 TClipboardFormatList.Clear Method

Not documented.

Pascal

```
procedure Clear;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormatList Class(see page 243)

10.1.4.3 TClipboardFormatList.Create Constructor

Not documented.

Pascal

```
constructor Create;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormatList Class(see page 243)

10.1.4.4 TClipboardFormatList.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormatList Class(see page 243)

10.1.4.5 EnumerateFormats

10.1.4.5.1 TClipboardFormatList.EnumerateFormats Method (TVirtualTreeClass, TFormatEtcArray, TClipboardFormats)

Returns a list of format records for the given class.

Pascal

```
procedure EnumerateFormats(TreeClass: TVirtualTreeClass; var Formats: TFormatEtcArray; const
AllowedFormats: TClipboardFormats = nil); overload;
```

Description

If assigned the AllowedFormats is used to limit the enumerated formats to those described in the list.

Class

TClipboardFormatList Class(see page 243)

10.1.4.5.2 TClipboardFormatList.EnumerateFormats Method (TVirtualTreeClass, TStrings)

Returns a list of format descriptions for the given class.

Pascal

```
procedure EnumerateFormats(TreeClass: TVirtualTreeClass; const Formats: TStrings); overload;
```

Class

TClipboardFormatList Class(see page 243)

10.1.4.6 FindFormat

10.1.4.6.1 TClipboardFormatList.FindFormat Method (Word, string)

Not documented.

Pascal

```
function FindFormat(Fmt: Word; var Description: string): TVirtualTreeClass; overload;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormatList Class(see page 243)

10.1.4.6.2 TClipboardFormatList.FindFormat Method (string)

Not documented.

Pascal

```
function FindFormat(FormatString: string): PClipboardFormatListEntry; overload;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormatList Class(see page 243)

10.1.4.6.3 TClipboardFormatList.FindFormat Method (string, Word)

Not documented.

Pascal

```
function FindFormat(FormatString: string; var Fmt: Word): TVirtualTreeClass; overload;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormatList Class(see page 243)

10.1.5 TClipboardFormats Class

List of strings describing clipboard formats.

Pascal

```
TClipboardFormats = class(TStringList);
```

Description

This class is an extended string list which allows to enter description strings for clipboard formats which are checked agains registered formats and only accepted if the particular format could be found. This way there is an unambiqious and portable description of allowed clipboard formats possible.

Group

Classes(see page 86)

Methods

Add(see TClipboardFormats.Add Method , page 247)
Adds a new format to the internal list.

Create(see TClipboardFormats.Create Constructor , page 247)
Constructor of the class.

Insert(see TClipboardFormats.Add Method , page 247)
Adds a new format to the internal list.

Properties

Owner(see TClipboardFormats.Owner Property, page 247)
Not documented.

Legend

public

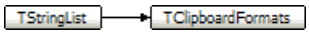
Property

read only

Method

virtual

Class Hierarchy



File

VirtualTrees

10.1.5.1 TClipboardFormats.Owner Property

Not documented.

Pascal

```
property Owner: TBaseVirtualTree;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormats Class(see page 246)

10.1.5.2 TClipboardFormats.Add Method

Adds a new format to the internal list.

Pascal

```
function Add(const S: string): Integer; override;
procedure Insert(Index: Integer; const S: string); override;
```

Description

Adds or inserts a new format to the internal list but restricts additions to the clipbard formats to only those which are registered with the owner tree or one of its ancestors.

Class

TClipboardFormats Class(see page 246)

10.1.5.3 TClipboardFormats.Create Constructor

Constructor of the class.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree); virtual;
```

Description

Create initializes the class.

Class

TClipboardFormats Class(see page 246)

10.1.6 TCriticalSection Class

Not documented.

Pascal

```
TCriticalSection = class(TObject);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Fields

FSection(see TCriticalSection.FSection Field, page 248)
Not documented.

Group

Classes(see page 86)

Methods

Create(see TCriticalSection.Create Constructor , page 249)
Not documented.

Destroy(see TCriticalSection.Destroy Destructor , page 249)
Not documented.

Enter(see TCriticalSection.Enter Method , page 249)
Not documented.

Leave(see TCriticalSection.Leave Method , page 249)
Not documented.

Legend

protected

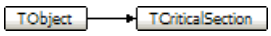
Data Member

public

Method

virtual

Class Hierarchy



File

VirtualTrees

10.1.6.1 TCriticalSection.FSection Field

Not documented.

Pascal

```
FSection: TRTLCriticalSection;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCriticalSection Class(see page 248)

10.1.6.2 TCriticalSection.Create Constructor

Not documented.

Pascal

```
constructor Create;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCriticalSection Class(see page 248)

10.1.6.3 TCriticalSection.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCriticalSection Class(see page 248)

10.1.6.4 TCriticalSection.Enter Method

Not documented.

Pascal

```
procedure Enter;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCriticalSection Class(see page 248)

10.1.6.5 TCriticalSection.Leave Method

Not documented.

Pascal

```
procedure Leave;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCriticalSection Class(see page 248)

10.1.7 TCustomStringTreeOptions Class

Enhanced options class for string trees.

Pascal

```
TCustomStringTreeOptions = class(TCustomVirtualTreeOptions);
```

Description

This class enhances the base class TCustomVirtualTreeOptions(see TCustomVirtualTreeOptions Class, page 308) by options related to a string tree.

Group

Classes(see page 86)

Methods

AssignTo(see TCustomStringTreeOptions.AssignTo Method , page 251)
Used to copy the options class.

Create(see TCustomStringTreeOptions.Create Constructor , page 251)
The constructor of the class.

TCustomVirtualTreeOptions Class

AssignTo(see TCustomVirtualTreeOptions.AssignTo Method , page 310)
Used to copy this option class to another option collection.

Create(see TCustomVirtualTreeOptions.Create Constructor , page 310)
Constructor of the class.

Properties

StringOptions(see TCustomStringTreeOptions.StringOptions Property, page 251)
The new options introduced by the class.

TCustomVirtualTreeOptions Class

AnimationOptions(see TCustomVirtualTreeOptions.AnimationOptions Property, page 309)
Options related to animations.

AutoOptions(see TCustomVirtualTreeOptions.AutoOptions Property, page 309)
Options related to automatic actions.

MiscOptions(see TCustomVirtualTreeOptions.MiscOptions Property, page 309)
Options not related to any other category.

Owner(see TCustomVirtualTreeOptions.Owner Property, page 309)
Owner tree to which the property class belongs.

PaintOptions(see TCustomVirtualTreeOptions.PaintOptions Property, page 310)
Options related to painting.

SelectionOptions(see TCustomVirtualTreeOptions.SelectionOptions Property, page 310)
Options related to the way nodes can be selected.

Legend

protected

Property

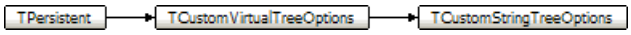
public

read only

Method

virtual

Class Hierarchy



File

VirtualTrees

10.1.7.1 TCustomStringTreeOptions.StringOptions Property

The new options introduced by the class.

Pascal

```
property StringOptions: TVTStringOptions;
```

Description

StringOptions provides access to the newly introduced options by which the base class is extended.

Class

TCustomStringTreeOptions Class(see page 250)

10.1.7.2 TCustomStringTreeOptions.AssignTo Method

Used to copy the options class.

Pascal

```
procedure AssignTo(Dest: TPersistent); override;
```

Description

You can either call this method directly or use the Assign method of the target class to do the assignment. Implementing AssignTo instead of Assign allows for future enhancements. TPersistent will automatically call AssignTo if there was no Assign method.

Class

TCustomStringTreeOptions Class(see page 250)

10.1.7.3 TCustomStringTreeOptions.Create Constructor

The constructor of the class.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree); override;
```

Description

The constructor initializes the class.

Class

TCustomStringTreeOptions Class(see page 250)

10.1.8 TCustomVirtualDrawTree Class

Simple owner draw descendant of the base tree.

Pascal

```
TCustomVirtualDrawTree = class(TBaseVirtualTree);
```

Description

TCustomVirtualDrawTree is a simple TBaseVirtualTree(see TBaseVirtualTree Class, page 88) descendant, which publishes the paint method through an event. This allows an application for self drawn tree views without overriding the base class.

Events

OnDrawHint(see TCustomVirtualDrawTree.OnDrawHint Event, page 271)
Triggered when a node hint or tooltip must be drawn.

OnDrawNode(see TCustomVirtualDrawTree.OnDrawNode Event, page 272)
Triggered when a node must be drawn.

OnGetHintSize(see TCustomVirtualDrawTree.OnGetHintSize Event, page 272)
Triggered when a node hint or tooltip is about to show.

OnGetNodeWidth(see TCustomVirtualDrawTree.OnGetNodeWidth Event, page 272)
Triggered when a node is about to be drawn.

TBaseVirtualTree Class

OnAdvancedHeaderDraw(see TBaseVirtualTree.OnAdvancedHeaderDraw Event, page 128)
Header paint support event.

OnAfterCellPaint(see TBaseVirtualTree.OnAfterCellPaint Event, page 128)
Paint support event.

OnAfterItemErase(see TBaseVirtualTree.OnAfterItemErase Event, page 129)
Paint support event.

OnAfterItemPaint(see TBaseVirtualTree.OnAfterItemPaint Event, page 129)
Paint support event.

OnAfterPaint(see TBaseVirtualTree.OnAfterPaint Event, page 129)
Paint support event.

OnBeforeCellPaint(see TBaseVirtualTree.OnBeforeCellPaint Event, page 130)
Paint support event.

OnBeforeItemErase(see TBaseVirtualTree.OnBeforeItemErase Event, page 130)
Paint support event.

OnBeforeItemPaint(see TBaseVirtualTree.OnBeforeItemPaint Event, page 130)
Paint support event.

OnBeforePaint(see TBaseVirtualTree.OnBeforePaint Event, page 131)
Paint support event.

OnChange(see TBaseVirtualTree.OnChange Event, page 131)
Navigation support event.

OnChecked(see TBaseVirtualTree.OnChecked Event, page 131)
Check support event.

OnChecking(see TBaseVirtualTree.OnChecking Event, page 131)
Check support event.

OnCollapsed(see TBaseVirtualTree.OnCollapsed Event, page 132)
Miscellaneous event.

OnCollapsing(see TBaseVirtualTree.OnCollapsing Event, page 132)
Miscellaneous event.

OnColumnClick(see TBaseVirtualTree.OnColumnClick Event, page 132)
Header and column support event.

OnColumnDblClick(see TBaseVirtualTree.OnColumnDblClick Event, page 132)
Header and column support event.

OnColumnResize(see TBaseVirtualTree.OnColumnResize Event, page 133)
Header and column support routine.

OnCompareNodes(see TBaseVirtualTree.OnCompareNodes Event, page 133)
Sort and search support event.

OnCreateDataObject(see TBaseVirtualTree.OnCreateDataObject Event, page 134)
Drag'n drop support event.

OnCreateDragManager(see TBaseVirtualTree.OnCreateDragManager Event, page 134)
Drag'n drop support event.

OnCreateEditor(see TBaseVirtualTree.OnCreateEditor Event, page 134)
Editing support event.

OnDragAllowed(see TBaseVirtualTree.OnDragAllowed Event, page 135)
Drag'n drop support event.

OnDragDrop(see TBaseVirtualTree.OnDragDrop Event, page 135)
Drag'n drop support event.

OnDragOver(see TBaseVirtualTree.OnDragOver Event, page 137)
Drag'n drop support event.

OnEditCancelled(see TBaseVirtualTree.OnEditCancelled Event, page 137)
Editing support event.

OnEdited(see TBaseVirtualTree.OnEdited Event, page 137)
Editing support event.

OnEditing(see TBaseVirtualTree.OnEditing Event, page 138)
Editing support event.

OnExpanded(see TBaseVirtualTree.OnExpanded Event, page 138)
Misscellaneous event.

OnExpanding(see TBaseVirtualTree.OnExpanding Event, page 138)
Miscellaneous event.

OnFocusChanged(see TBaseVirtualTree.OnFocusChanged Event, page 138)
Navigation support event.

OnFocusChanging(see TBaseVirtualTree.OnFocusChanging Event, page 139)
Navigation support event.

OnFreeNode(see TBaseVirtualTree.OnFreeNode Event, page 139)
Data management node.

OnGetCellIsEmpty(see TBaseVirtualTree.OnGetCellIsEmpty Event, page 139)
Triggered when the tree control needs to know whether a given column is empty.

OnGetCursor(see TBaseVirtualTree.OnGetCursor Event, page 140)
Miscellaneous event.

OnGetHeaderCursor(see TBaseVirtualTree.OnGetHeaderCursor Event, page 140)
Header and column support event.

OnGetHelpContext(see TBaseVirtualTree.OnGetHelpContext Event, page 140)
Miscellaneous event.

OnGetImageIndex(see TBaseVirtualTree.OnGetImageIndex Event, page 140)

Display management event.

OnGetImageIndexEx(see TBaseVirtualTree.OnGetImageIndexEx Event, page 141)
Not documented.

OnGetLineStyle(see TBaseVirtualTree.OnGetLineStyle Event, page 141)
Display management event.

OnGetNodeDataSize(see TBaseVirtualTree.OnGetNodeDataSize Event, page 142)
Data management event.

OnGetPopupMenu(see TBaseVirtualTree.OnGetPopupMenu Event, page 142)
Miscellaneous event.

OnGetUserClipboardFormats(see TBaseVirtualTree.OnGetUserClipboardFormats Event, page 142)
Drag'n drop and clipboard support event.

OnHeaderClick(see TBaseVirtualTree.OnHeaderClick Event, page 143)
Header & column support event.

OnHeaderDblClick(see TBaseVirtualTree.OnHeaderDblClick Event, page 143)
Header & column support event.

OnHeaderDragged(see TBaseVirtualTree.OnHeaderDragged Event, page 143)
Header & column support event.

OnHeaderDraggedOut(see TBaseVirtualTree.OnHeaderDraggedOut Event, page 144)
Header & column support event.

OnHeaderDragging(see TBaseVirtualTree.OnHeaderDragging Event, page 144)
Header & column support event.

OnHeaderDraw(see TBaseVirtualTree.OnHeaderDraw Event, page 144)
Header & column support event.

OnHeaderDrawQueryElements(see TBaseVirtualTree.OnHeaderDrawQueryElements Event, page 145)
Header & column support event.

OnHeaderMouseDown(see TBaseVirtualTree.OnHeaderMouseDown Event, page 145)
Header & column support event.

OnHeaderMouseMove(see TBaseVirtualTree.OnHeaderMouseMove Event, page 145)
Header & column support event.

OnHeaderMouseUp(see TBaseVirtualTree.OnHeaderMouseUp Event, page 145)
Header & column support event.

OnHotChange(see TBaseVirtualTree.OnHotChange Event, page 146)
Navigation support event.

OnIncrementalSearch(see TBaseVirtualTree.OnIncrementalSearch Event, page 146)
Miscellaneous event.

OnInitChildren(see TBaseVirtualTree.OnInitChildren Event, page 147)
Node management event.

OnInitNode(see TBaseVirtualTree.OnInitNode Event, page 147)
Node management event.

OnKeyAction(see TBaseVirtualTree.OnKeyAction Event, page 148)
Miscellaneous event.

OnLoadNode(see TBaseVirtualTree.OnLoadNode Event, page 148)
Streaming support event.

OnMeasureItem(see TBaseVirtualTree.OnMeasureItem Event, page 148)
Miscellaneous event.

OnNodeCopied(see TBaseVirtualTree.OnNodeCopied Event, page 149)
Miscellaneous event.

OnNodeCopying(see TBaseVirtualTree.OnNodeCopying Event, page 149)
Miscellaneous event.

OnNodeMoved(see TBaseVirtualTree.OnNodeMoved Event, page 149)
Miscellaneous event.

OnNodeMoving(see TBaseVirtualTree.OnNodeMoving Event, page 150)
Miscellaneous event.

OnPaintBackground(see TBaseVirtualTree.OnPaintBackground Event, page 150)
Paint support event.

OnRenderOLEData(see TBaseVirtualTree.OnRenderOLEData Event, page 150)
Drag'n drop and clipboard support event.

OnResetNode(see TBaseVirtualTree.OnResetNode Event, page 151)
Node management event.

OnSaveNode(see TBaseVirtualTree.OnSaveNode Event, page 151)
Streaming support event.

OnScroll(see TBaseVirtualTree.OnScroll Event, page 152)
Miscellaneous event.

OnShowScrollbar(see TBaseVirtualTree.OnShowScrollbar Event, page 152)
Not documented.

OnStateChange(see TBaseVirtualTree.OnStateChange Event, page 152)
Miscellaneous event.

OnStructureChange(see TBaseVirtualTree.OnStructureChange Event, page 152)
Miscellaneous event.

OnUpdating(see TBaseVirtualTree.OnUpdating Event, page 153)
Miscellaneous event.

Group

Classes(see page 86)

Methods

DoDrawHint(see TCustomVirtualDrawTree.DoDrawHint Method , page 272)
Overridable method which triggers OnDrawHint(see TCustomVirtualDrawTree.OnDrawHint Event, page 271).

DoGetHintSize(see TCustomVirtualDrawTree.DoGetHintSize Method , page 273)
Overridable method which triggers OnGetHintSize(see TCustomVirtualDrawTree.OnGetHintSize Event, page 272).

DoGetNodeWidth(see TCustomVirtualDrawTree.DoGetNodeWidth Method , page 273)
Overridable method which triggers OnGetNodeWidth(see TCustomVirtualDrawTree.OnGetNodeWidth Event, page 272).

DoPaintNode(see TCustomVirtualDrawTree.DoPaintNode Method , page 273)
Overridable method which triggers OnPaintNode.

TBaseVirtualTree Class

AbsoluteIndex(see TBaseVirtualTree.AbsoluteIndex Method , page 160)
Reads the overall index of a node.

AddChild(see TBaseVirtualTree.AddChild Method , page 160)
Creates and adds a new child node to given node.

AddFromStream(see TBaseVirtualTree.AddFromStream Method , page 161)
Adds the content from the given stream to the given node.

AddToSelection(see TBaseVirtualTree.AddToSelection Method (PVirtualNode), page 161)
Adds one or more nodes to the current selection.

AdjustPaintCellRect(see TBaseVirtualTree.AdjustPaintCellRect Method , page 162)
Used in descentants to modify the clip rectangle of the current column while painting a certain node.

AdjustPanningCursor(see TBaseVirtualTree.AdjustPanningCursor Method , page 162)
Loads the proper cursor which indicates into which direction scrolling is done.

AdviseChangeEvent(see TBaseVirtualTree.AdviseChangeEvent Method , page 162)
Used to register a delayed change event.

AllocateInternalDataArea(see TBaseVirtualTree.AllocateInternalDataArea Method , page 163)
Registration method to allocate tree internal data per node.

Animate(see TBaseVirtualTree.Animate Method , page 163)
Support method for animated actions in the tree view.

Assign(see TBaseVirtualTree.Assign Method , page 164)
Used to copy properties from another Virtual Treeview.

BeginDrag(see TBaseVirtualTree.BeginDrag Method , page 164)
Starts an OLE drag'n drop operation.

BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164)
Enters the tree into a special synchronized mode.

BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164)
Locks the tree view to perform several update operations.

CalculateSelectionRect(see TBaseVirtualTree.CalculateSelectionRect Method , page 165)
Support method for draw selection.

CanAutoScroll(see TBaseVirtualTree.CanAutoScroll Method , page 165)
Determines whether the tree can currently auto scroll its window.

CancelCutOrCopy(see TBaseVirtualTree.CancelCutOrCopy Method , page 165)
Canceles any pending cut or copy clipboard operation.

CancelEditNode(see TBaseVirtualTree.CancelEditNode Method , page 166)
Cancel the current edit operation, if there is any.

CanEdit(see TBaseVirtualTree.CanEdit Method , page 166)
Determines whether a node can be edited or not.

CanFocus(see TBaseVirtualTree.CanFocus Method , page 166)
Support method to determine whether the tree window can receive the input focus.

CanShowDragImage(see TBaseVirtualTree.CanShowDragImage Method , page 166)
Determines whether a drag image should be shown.

Change(see TBaseVirtualTree.Change Method , page 167)
Central method called when a node's selection state changes.

ChangeScale(see TBaseVirtualTree.ChangeScale Method , page 167)
Helper method called by the VCL when control resizing is due.

CheckParentCheckState(see TBaseVirtualTree.CheckParentCheckState Method , page 167)
Helper method for recursive check state changes.

Clear(see TBaseVirtualTree.Clear Method , page 168)
Clears the tree and removes all nodes.

ClearChecked(see TBaseVirtualTree.ClearChecked Method , page 168)
Not documented.

ClearSelection(see TBaseVirtualTree.ClearSelection Method , page 168)
Removes all nodes from the current selection.

ClearTempCache(see TBaseVirtualTree.ClearTempCache Method , page 168)
Helper method to clear(see TBaseVirtualTree.Clear Method , page 168) the internal temporary node cache.

ColumnIsEmpty(see TBaseVirtualTree.ColumnIsEmpty Method , page 169)
Used to determine if a cell is considered as being empty.

CopyTo(see TBaseVirtualTree.CopyTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 169)
Copies Source and all its child nodes to Target.

CopyToClipBoard(see TBaseVirtualTree.CopyToClipBoard Method , page 169)
Copies all currently selected nodes to the clipboard.

CountLevelDifference(see TBaseVirtualTree.CountLevelDifference Method , page 170)
Determines the level difference of two nodes.

CountVisibleChildren(see TBaseVirtualTree.CountVisibleChildren Method , page 170)
Determines the number of visible child nodes of the given node.

Create(see TBaseVirtualTree.Create Constructor , page 170)
Constructor of the control

CreateParams(see TBaseVirtualTree.CreateParams Method , page 171)
Prepares the creation of the controls window handle.

CreateWnd(see TBaseVirtualTree.CreateWnd Method , page 171)

Initializes data, which depends on the window handle.

CutToClipBoard(see TBaseVirtualTree.CutToClipBoard Method , page 171)
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties(see TBaseVirtualTree.DefineProperties Method , page 171)
Helper method to customize loading and saving persistent tree data.

DeleteChildren(see TBaseVirtualTree.DeleteChildren Method , page 172)
Removes all child nodes from the given node.

DeleteNode(see TBaseVirtualTree.DeleteNode Method , page 172)
Removes the given node from the tree.

DeleteSelectedNodes(see TBaseVirtualTree.DeleteSelectedNodes Method , page 172)
Removes all currently selected nodes form the tree.

Destroy(see TBaseVirtualTree.Destroy Destructor , page 173)
Destructor of the control.

DetermineHiddenChildrenFlag(see TBaseVirtualTree.DetermineHiddenChildrenFlag Method , page 173)
Determines whether all children of a given node are hidden.

DetermineHiddenChildrenFlagAllNodes(see TBaseVirtualTree.DetermineHiddenChildrenFlagAllNodes Method , page 173)
Determines whether all children of all nodes are hidden.

DetermineHitPositionLTR(see TBaseVirtualTree.DetermineHitPositionLTR Method , page 174)
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineHitPositionRTL(see TBaseVirtualTree.DetermineHitPositionLTR Method , page 174)
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineNextCheckState(see TBaseVirtualTree.DetermineNextCheckState Method , page 174)
Not documented.

DetermineScrollDirections(see TBaseVirtualTree.DetermineScrollDirections Method , page 174)
Not documented.

DoAdvancedHeaderDraw(see TBaseVirtualTree.DoAdvancedHeaderDraw Method , page 174)
Not documented.

DoAfterCellPaint(see TBaseVirtualTree.DoAfterCellPaint Method , page 175)
Not documented.

DoAfterItemErase(see TBaseVirtualTree.DoAfterItemErase Method , page 175)
Not documented.

DoAfterItemPaint(see TBaseVirtualTree.DoAfterItemPaint Method , page 175)
Not documented.

DoAfterPaint(see TBaseVirtualTree.DoAfterPaint Method , page 175)
Not documented.

DoAutoScroll(see TBaseVirtualTree.DoAutoScroll Method , page 176)
Enables or disables the auto scroll timer.

DoBeforeCellPaint(see TBaseVirtualTree.DoBeforeCellPaint Method , page 176)
Not documented.

DoBeforeDrag(see TBaseVirtualTree.DoBeforeDrag Method , page 176)
Not documented.

DoBeforeItemErase(see TBaseVirtualTree.DoBeforeItemErase Method , page 177)
Not documented.

DoBeforeItemPaint(see TBaseVirtualTree.DoBeforeItemPaint Method , page 177)
Not documented.

DoBeforePaint(see TBaseVirtualTree.DoBeforePaint Method , page 177)
Not documented.

DoCancelEdit(see TBaseVirtualTree.DoCancelEdit Method , page 177)
Called when the tree should stop editing without accepting changed values.

DoCanEdit(see TBaseVirtualTree.DoCanEdit Method , page 178)
Not documented.

DoChange(see TBaseVirtualTree.DoChange Method , page 178)
Not documented.

DoCheckClick(see TBaseVirtualTree.DoCheckClick Method , page 178)
Not documented.

DoChecked(see TBaseVirtualTree.DoChecked Method , page 178)
Not documented.

DoChecking(see TBaseVirtualTree.DoChecking Method , page 179)
Not documented.

DoCollapsed(see TBaseVirtualTree.DoCollapsed Method , page 179)
Not documented.

DoCollapsing(see TBaseVirtualTree.DoCollapsing Method , page 179)
Not documented.

DoColumnClick(see TBaseVirtualTree.DoColumnClick Method , page 179)
Not documented.

DoColumnDblClick(see TBaseVirtualTree.DoColumnDblClick Method , page 180)
Not documented.

DoColumnResize(see TBaseVirtualTree.DoColumnResize Method , page 180)
Not documented.

DoCompare(see TBaseVirtualTree.DoCompare Method , page 180)
Not documented.

DoCreateDataObject(see TBaseVirtualTree.DoCreateDataObject Method , page 180)
Not documented.

DoCreateDragManager(see TBaseVirtualTree.DoCreateDragManager Method , page 181)
Not documented.

DoCreateEditor(see TBaseVirtualTree.DoCreateEditor Method , page 181)
Not documented.

DoDragDrop(see TBaseVirtualTree.DoDragDrop Method , page 181)
Not documented.

DoDragExpand(see TBaseVirtualTree.DoDragExpand Method , page 181)
Not documented.

DoDragging(see TBaseVirtualTree.DoDragging Method , page 182)
Internal method which handles drag' drop.

DoDragOver(see TBaseVirtualTree.DoDragOver Method , page 182)
Not documented.

DoEdit(see TBaseVirtualTree.DoEdit Method , page 182)
Initiates editing of the currently set focused column and edit node.

DoEndDrag(see TBaseVirtualTree.DoEndDrag Method , page 182)
Not documented.

DoEndEdit(see TBaseVirtualTree.DoEndEdit Method , page 183)
Stops the current edit operation and takes over the new content.

DoExpanded(see TBaseVirtualTree.DoExpanded Method , page 183)
Not documented.

DoExpanding(see TBaseVirtualTree.DoExpanding Method , page 183)
Not documented.

DoFocusChange(see TBaseVirtualTree.DoFocusChange Method , page 184)
Not documented.

DoFocusChanging(see TBaseVirtualTree.DoFocusChanging Method , page 184)
Not documented.

DoFocusNode(see TBaseVirtualTree.DoFocusNode Method , page 184)
Internal method to set the focused node.

DoFreeNode(see TBaseVirtualTree.DoFreeNode Method , page 184)

Not documented.

DoGetAnimationType(see TBaseVirtualTree.DoGetAnimationType Method , page 185)
Determines the type of animation to be used.

DoGetCursor(see TBaseVirtualTree.DoGetCursor Method , page 185)
Not documented.

DoGetHeaderCursor(see TBaseVirtualTree.DoGetHeaderCursor Method , page 185)
Not documented.

DoGetImageIndex(see TBaseVirtualTree.DoGetImageIndex Method , page 186)
Not documented.

DoGetLineStyle(see TBaseVirtualTree.DoGetLineStyle Method , page 186)
Not documented.

DoGetNodeHint(see TBaseVirtualTree.DoGetNodeHint Method , page 186)
Not documented.

DoGetNodeTooltip(see TBaseVirtualTree.DoGetNodeTooltip Method , page 186)
Not documented.

DoGetNodeWidth(see TBaseVirtualTree.DoGetNodeWidth Method , page 187)
Overridable method which always retuns 0.

DoGetPopupMenu(see TBaseVirtualTree.DoGetPopupMenu Method , page 187)
Overridable method which triggers the OnGetPopup event.

DoGetUserClipboardFormats(see TBaseVirtualTree.DoGetUserClipboardFormats Method , page 187)
Not documented.

DoHeaderClick(see TBaseVirtualTree.DoHeaderClick Method , page 187)
Not documented.

DoHeaderDblClick(see TBaseVirtualTree.DoHeaderDblClick Method , page 188)
Not documented.

DoHeaderDragged(see TBaseVirtualTree.DoHeaderDragged Method , page 188)
Not documented.

DoHeaderDraggedOut(see TBaseVirtualTree.DoHeaderDraggedOut Method , page 188)
Not documented.

DoHeaderDragging(see TBaseVirtualTree.DoHeaderDragging Method , page 189)
Not documented.

DoHeaderDraw(see TBaseVirtualTree.DoHeaderDraw Method , page 189)
Not documented.

DoHeaderDrawQueryElements(see TBaseVirtualTree.DoHeaderDrawQueryElements Method , page 189)
Not documented.

DoHeaderMouseDown(see TBaseVirtualTree.DoHeaderMouseDown Method , page 189)
Not documented.

DoHeaderMouseMove(see TBaseVirtualTree.DoHeaderMouseMove Method , page 190)
Not documented.

DoHeaderMouseUp(see TBaseVirtualTree.DoHeaderMouseUp Method , page 190)
Not documented.

DoHotChange(see TBaseVirtualTree.DoHotChange Method , page 190)
Not documented.

DoIncrementalSearch(see TBaseVirtualTree.DoIncrementalSearch Method , page 190)
Not documented.

DoInitChildren(see TBaseVirtualTree.DoInitChildren Method , page 191)
Not documented.

DoInitNode(see TBaseVirtualTree.DoInitNode Method , page 191)
Not documented.

DoKeyAction(see TBaseVirtualTree.DoKeyAction Method , page 191)
Not documented.

DoLoadUserData(see TBaseVirtualTree.DoLoadUserData Method , page 191)
Not documented.

DoMeasureItem(see TBaseVirtualTree.DoMeasureItem Method , page 192)
Not documented.

DoNodeCopied(see TBaseVirtualTree.DoNodeCopied Method , page 192)
Not documented.

DoNodeCopying(see TBaseVirtualTree.DoNodeCopying Method , page 192)
Not documented.

DoNodeMoved(see TBaseVirtualTree.DoNodeMoved Method , page 192)
Not documented.

DoNodeMoving(see TBaseVirtualTree.DoNodeMoving Method , page 193)
Not documented.

DoPaintBackground(see TBaseVirtualTree.DoPaintBackground Method , page 193)
Not documented.

DoPaintDropMark(see TBaseVirtualTree.DoPaintDropMark Method , page 193)
Overridable method which draws the small line on top of a nodes image depending on the current drop state.

DoPaintNode(see TBaseVirtualTree.DoPaintNode Method , page 193)
Overridable method which does nothing.

DoPopupMenu(see TBaseVirtualTree.DoPopupMenu Method , page 194)
Overridable method which shows the popup menu for the given node.

DoRenderOLEData(see TBaseVirtualTree.DoRenderOLEData Method , page 194)
Not documented.

DoReset(see TBaseVirtualTree.DoReset Method , page 194)
Not documented.

DoSaveUserData(see TBaseVirtualTree.DoSaveUserData Method , page 194)
Not documented.

DoScroll(see TBaseVirtualTree.DoScroll Method , page 195)
Overridable method which triggers the OnScroll(see TBaseVirtualTree.OnScroll Event, page 152) event.

DoSetOffsetXY(see TBaseVirtualTree.DoSetOffsetXY Method , page 195)
Internal core routine to set the tree's scroll position.

DoShowScrollbar(see TBaseVirtualTree.DoShowScrollbar Method , page 195)
Not documented.

DoStartDrag(see TBaseVirtualTree.DoStartDrag Method , page 196)
Not documented.

DoStateChange(see TBaseVirtualTree.DoStateChange Method , page 196)
Not documented.

DoStructureChange(see TBaseVirtualTree.DoStructureChange Method , page 196)
Not documented.

DoTimerScroll(see TBaseVirtualTree.DoTimerScroll Method , page 196)
Callback method which is triggered whenever the scroll timer fires.

DoUpdating(see TBaseVirtualTree.DoUpdating Method , page 197)
Not documented.

DoValidateCache(see TBaseVirtualTree.DoValidateCache Method , page 197)
Not documented.

DragCanceled(see TBaseVirtualTree.DragCanceled Method , page 197)
Called by the VCL when a drag'n drop operation was canceled by the user.

DragDrop(see TBaseVirtualTree.DragDrop Method , page 197)
Helper method, which is used when a drag operation is finished.

DragEnter(see TBaseVirtualTree.DragEnter Method , page 198)
Not documented.

DragFinished(see TBaseVirtualTree.DragFinished Method , page 198)

Called when a drag operation is finished (accepted or cancelled).

Dragging(see TBaseVirtualTree.Dragging Method , page 198)
Returns true if a drag'n drop operation is in progress.

DragLeave(see TBaseVirtualTree.DragLeave Method , page 199)
Not documented.

DragOver(see TBaseVirtualTree.DragOver Method , page 199)
Not documented.

DrawDottedHLine(see TBaseVirtualTree.DrawDottedHLine Method , page 199)
Not documented.

DrawDottedVLine(see TBaseVirtualTree.DrawDottedVLine Method , page 199)
Not documented.

EditNode(see TBaseVirtualTree.EditNode Method , page 200)
Starts editing the given node if allowed to.

EndEditNode(see TBaseVirtualTree.EndEditNode Method , page 200)
Stops node editing if it was started before.

EndSynch(see TBaseVirtualTree.EndSynch Method , page 200)
Counterpart to BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164).

EndUpdate(see TBaseVirtualTree.EndUpdate Method , page 201)
Resets the update lock set by BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164).

ExecuteAction(see TBaseVirtualTree.ExecuteAction Method , page 201)
Not documented.

FindNodeInSelection(see TBaseVirtualTree.FindNodeInSelection Method , page 201)
Helper method to find the given node in the current selection.

FinishChunkHeader(see TBaseVirtualTree.FinishChunkHeader Method , page 201)
Not documented.

FinishCutOrCopy(see TBaseVirtualTree.FinishCutOrCopy Method , page 202)
Stops any pending cut or copy clipboard operation.

FlushClipboard(see TBaseVirtualTree.FlushClipboard Method , page 202)
Renders all pending clipboard data.

FontChanged(see TBaseVirtualTree.FontChanged Method , page 202)
Not documented.

FullCollapse(see TBaseVirtualTree.FullCollapse Method , page 203)
Collapses all nodes in the tree.

FullExpand(see TBaseVirtualTree.FullExpand Method , page 203)
Expands all nodes in the tree.

GetBorderDimensions(see TBaseVirtualTree.GetBorderDimensions Method , page 203)
Not documented.

GetCheckImage(see TBaseVirtualTree.GetCheckImage Method , page 203)
Not documented.

GetCheckImageListFor(see TBaseVirtualTree.GetCheckImageListFor Method , page 204)
Not documented.

GetColumnClass(see TBaseVirtualTree.GetColumnClass Method , page 204)
Returns the class to be used to manage columns in the tree.

GetControlsAlignment(see TBaseVirtualTree.GetControlsAlignment Method , page 204)
Not documented.

GetDisplayRect(see TBaseVirtualTree.GetDisplayRect Method , page 205)
Returns the visible region used by the given node in client coordinates.

GetFirst(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstChecked(see TBaseVirtualTree.GetFirstChecked Method , page 206)
Not documented.

GetFirstChild(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstCutCopy(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstInitialized(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstSelected(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisible(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleChild(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleChildNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetHeaderClass(see TBaseVirtualTree.GetHeaderClass Method , page 206)
Returns the header class to be used by the tree.

GetHintWindowClass(see TBaseVirtualTree.GetHintWindowClass Method , page 206)
Not documented.

GetHitTestInfoAt(see TBaseVirtualTree.GetHitTestInfoAt Method , page 206)
Returns information about the node at the given position.

GetImageIndex(see TBaseVirtualTree.GetImageIndex Method , page 207)
Not documented.

GetLast(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastChild(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastChildNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastInitialized(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisible(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleChild(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleChildNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetMaxColumnWidth(see TBaseVirtualTree.GetMaxColumnWidth Method , page 208)
Returns the width of the largest node in the given column.

GetMaxRightExtend(see TBaseVirtualTree.GetMaxRightExtend Method , page 208)
Determines the maximum with of the currently visible part of the tree.

GetNativeClipboardFormats(see TBaseVirtualTree.GetNativeClipboardFormats Method , page 208)
Used to let descendants and the application add their own supported clipboard formats.

GetNext(see TBaseVirtualTree.GetNext Method , page 209)

Group of node navigation functions.

GetNextChecked(see TBaseVirtualTree.GetNextChecked Method , page 209)
Not documented.

GetNextCutCopy(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextInitialized(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextSelected(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextSibling(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisible(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleSibling(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleSiblingNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNodeAt(see TBaseVirtualTree.GetNodeAt Method (Integer, Integer), page 210)
Not documented.

GetNodeData(see TBaseVirtualTree.GetNodeData Method , page 210)
Returns the address of the user data area of the given node.

GetNodeLevel(see TBaseVirtualTree.GetNodeLevel Method , page 210)
Returns the indentation level of the given node.

GetOptionsClass(see TBaseVirtualTree.GetOptionsClass Method , page 211)
Customization helper to determine which options class the tree should use.

GetPrevious(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousInitialized(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousSibling(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisible(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleSibling(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleSiblingNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetSortedCutCopySet(see TBaseVirtualTree.GetSortedCutCopySet Method , page 212)
Returns a sorted list of nodes, which are marked for s cut or copy clipboard operation.

GetSortedSelection(see TBaseVirtualTree.GetSortedSelection Method , page 213)
Returns a sorted list of all currently selected nodes.

GetTextInfo(see TBaseVirtualTree.GetTextInfo Method , page 213)
Helper method for node editors, hints etc.

GetTreeFromDataObject(see TBaseVirtualTree.GetTreeFromDataObject Method , page 213)
OLE drag'n drop and clipboard support method.

GetTreeRect(see TBaseVirtualTree.GetTreeRect Method , page 214)
Returns the size of the virtual tree image.

GetVisibleParent(see TBaseVirtualTree.GetVisibleParent Method , page 214)
Returns the first (nearest) parent node, which is visible.

HandleHotTrack(see TBaseVirtualTree.HandleHotTrack Method , page 214)
Not documented.

HandleIncrementalSearch(see TBaseVirtualTree.HandleIncrementalSearch Method , page 214)
Not documented.

HandleMouseDblClick(see TBaseVirtualTree.HandleMouseDblClick Method , page 215)
Not documented.

HandleMouseDown(see TBaseVirtualTree.HandleMouseDown Method , page 215)
Not documented.

HandleMouseUp(see TBaseVirtualTree.HandleMouseUp Method , page 215)
Not documented.

HasAsParent(see TBaseVirtualTree.HasAsParent Method , page 215)
Determines if the given node has got another node as one of its parents.

HasImage(see TBaseVirtualTree.HasImage Method , page 216)
Not documented.

HasPopupMenu(see TBaseVirtualTree.HasPopupMenu Method , page 216)
Determines whether there is a pop up menu assigned to the tree.

InitChildren(see TBaseVirtualTree.InitChildren Method , page 216)
Not documented.

InitNode(see TBaseVirtualTree.InitNode Method , page 217)
Not documented.

InsertNode(see TBaseVirtualTree.InsertNode Method , page 217)
Inserts a new node and returns it to the caller.

InternalAddFromStream(see TBaseVirtualTree.InternalAddFromStream Method , page 217)
Not documented.

InternalAddToSelection(see TBaseVirtualTree.InternalAddToSelection Method (PVirtualNode, Boolean), page 218)
Not documented.

InternalCacheNode(see TBaseVirtualTree.InternalCacheNode Method , page 218)
Not documented.

InternalClearSelection(see TBaseVirtualTree.InternalClearSelection Method , page 218)
Not documented.

InternalConnectNode(see TBaseVirtualTree.InternalConnectNode Method , page 219)
Not documented.

InternalData(see TBaseVirtualTree.InternalData Method , page 219)
Returns the address of the internal data for a tree class.

InternalDisconnectNode(see TBaseVirtualTree.InternalDisconnectNode Method , page 219)
Not documented.

InternalRemoveFromSelection(see TBaseVirtualTree.InternalRemoveFromSelection Method , page 220)
Not documented.

InvalidateCache(see TBaseVirtualTree.InvalidateCache Method , page 220)
Empties the internal node cache and marks it as invalid.

InvalidateChildren(see TBaseVirtualTree.InvalidateChildren Method , page 220)
Invalidates all children of the given node.

InvalidateColumn(see TBaseVirtualTree.InvalidateColumn Method , page 220)
Invalidates the client area part of a column.

InvalidateNode(see TBaseVirtualTree.InvalidateNode Method , page 221)

Invalidates the given node.

InvalidateToBottom(see TBaseVirtualTree.InvalidateToBottom Method , page 221)
Invalidates the client area starting with the top position of the given node.

InvertSelection(see TBaseVirtualTree.InvertSelection Method , page 221)
Inverts the current selection.

IsEditing(see TBaseVirtualTree.IsEditing Method , page 222)
Tells the caller whether the tree is currently in edit mode.

IsMouseSelecting(see TBaseVirtualTree.IsMouseSelecting Method , page 222)
Tell the caller whether the tree is currently in draw selection mode.

IterateSubtree(see TBaseVirtualTree.IterateSubtree Method , page 222)
Iterator method to go through all nodes of a given sub tree.

Loaded(see TBaseVirtualTree.Loaded Method , page 223)
Not documented.

LoadFromFile(see TBaseVirtualTree.LoadFromFile Method , page 223)
Loads previously streamed out tree data back in again.

LoadFromStream(see TBaseVirtualTree.LoadFromFile Method , page 223)
Loads previously streamed out tree data back in again.

MainColumnChanged(see TBaseVirtualTree.MainColumnChanged Method , page 223)
Not documented.

MarkCutCopyNodes(see TBaseVirtualTree.MarkCutCopyNodes Method , page 223)
Not documented.

MeasureItemHeight(see TBaseVirtualTree.MeasureItemHeight Method , page 224)
Not documented.

MouseMove(see TBaseVirtualTree.MouseMove Method , page 224)
Not documented.

MoveTo(see TBaseVirtualTree.MoveTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 224)
Moves Source and all its child nodes to Target.

Notification(see TBaseVirtualTree.Notification Method , page 225)
Not documented.

OriginalWMNCPaint(see TBaseVirtualTree.OriginalWMNCPaint Method , page 225)
Not documented.

Paint(see TBaseVirtualTree.Paint Method , page 225)
TControl's Paint method used here to display the tree.

PaintCheckImage(see TBaseVirtualTree.PaintCheckImage Method , page 225)
Not documented.

PaintImage(see TBaseVirtualTree.PaintImage Method , page 226)
Not documented.

PaintNodeButton(see TBaseVirtualTree.PaintNodeButton Method , page 226)
Not documented.

PaintSelectionRectangle(see TBaseVirtualTree.PaintSelectionRectangle Method , page 226)
Not documented.

PaintTree(see TBaseVirtualTree.PaintTree Method , page 226)
Main paint routine for the tree image.

PaintTreeLines(see TBaseVirtualTree.PaintTreeLines Method , page 227)
Not documented.

PanningWindowProc(see TBaseVirtualTree.PanningWindowProc Method , page 227)
Not documented.

PasteFromClipboard(see TBaseVirtualTree.PasteFromClipboard Method , page 227)
Inserts the content of the clipboard into the tree.

PrepareDragImage(see TBaseVirtualTree.PrepareDragImage Method , page 228)
Not documented.

Print(see TBaseVirtualTree.Print Method , page 228)
Not documented.

ProcessDrop(see TBaseVirtualTree.ProcessDrop Method , page 228)
Helper method to ease OLE drag'n drop operations.

ProcessOLEData(see TBaseVirtualTree.ProcessOLEData Method , page 229)
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk(see TBaseVirtualTree.ReadChunk Method , page 229)
Not documented.

ReadNode(see TBaseVirtualTree.ReadNode Method , page 229)
Not documented.

RedirectFontChangeEvent(see TBaseVirtualTree.RedirectFontChangeEvent Method , page 230)
Not documented.

ReinitChildren(see TBaseVirtualTree.ReinitChildren Method , page 230)
Forces all child nodes of Node to be reinitialized.

ReinitNode(see TBaseVirtualTree.ReinitNode Method , page 230)
Forces a reinitialization of the given node.

RemoveFromSelection(see TBaseVirtualTree.RemoveFromSelection Method , page 230)
Removes the given node from the current selection.

RenderOLEData(see TBaseVirtualTree.RenderOLEData Method , page 231)
Renders pending OLE data.

RepaintNode(see TBaseVirtualTree.RepaintNode Method , page 231)
Causes the treeview to repaint the given node.

ResetNode(see TBaseVirtualTree.ResetNode Method , page 231)
Resets the given node to uninitialized.

ResetRangeAnchor(see TBaseVirtualTree.ResetRangeAnchor Method , page 232)
Not documented.

RestoreFontChangeEvent(see TBaseVirtualTree.RestoreFontChangeEvent Method , page 232)
Not documented.

SaveToFile(see TBaseVirtualTree.SaveToFile Method , page 232)
Saves the entire content of the tree into a file or stream.

SaveToStream(see TBaseVirtualTree.SaveToFile Method , page 232)
Saves the entire content of the tree into a file or stream.

ScrollIntoView(see TBaseVirtualTree.ScrollIntoView Method , page 232)
Scrolls the tree so that the given node comes in the client area.

SelectAll(see TBaseVirtualTree.SelectAll Method , page 233)
Selects all nodes in the tree.

SelectNodes(see TBaseVirtualTree.SelectNodes Method , page 233)
Selects a range of nodes.

SetBiDiMode(see TBaseVirtualTree.SetBiDiMode Method , page 233)
Not documented.

SetFocusedNodeAndColumn(see TBaseVirtualTree.SetFocusedNodeAndColumn Method , page 234)
Not documented.

SkipNode(see TBaseVirtualTree.SkipNode Method , page 234)
Not documented.

Sort(see TBaseVirtualTree.Sort Method , page 234)
Sorts the given node.

SortTree(see TBaseVirtualTree.SortTree Method , page 234)
Sorts the entire tree view.

StartWheelPanning(see TBaseVirtualTree.StartWheelPanning Method , page 235)
Not documented.

StopWheelPanning(see TBaseVirtualTree.StopWheelPanning Method , page 235)

Not documented.

StructureChange(see TBaseVirtualTree.StructureChange Method , page 235)
Not documented.

SuggestDropEffect(see TBaseVirtualTree.SuggestDropEffect Method , page 236)
Not documented.

ToggleNode(see TBaseVirtualTree.ToggleNode Method , page 236)
Changes a node's expand state to the opposite state.

ToggleSelection(see TBaseVirtualTree.ToggleSelection Method , page 236)
Toggles the selection state of a range of nodes.

UnselectNodes(see TBaseVirtualTree.UnselectNodes Method , page 236)
Deselects a range of nodes.

UpdateAction(see TBaseVirtualTree.UpdateAction Method , page 237)
Not documented.

UpdateDesigner(see TBaseVirtualTree.UpdateDesigner Method , page 237)
Not documented.

UpdateEditBounds(see TBaseVirtualTree.UpdateEditBounds Method , page 237)
Not documented.

UpdateHeaderRect(see TBaseVirtualTree.UpdateHeaderRect Method , page 237)
Not documented.

UpdateHorizontalScrollBar(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateScrollBars(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateVerticalScrollBar(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateWindowAndDragImage(see TBaseVirtualTree.UpdateWindowAndDragImage Method , page 238)
Not documented.

UseRightToLeftReading(see TBaseVirtualTree.UseRightToLeftReading Method , page 238)
Helper method for right-to-left layout.

ValidateCache(see TBaseVirtualTree.ValidateCache Method , page 239)
Initiates the validation of the internal node cache.

ValidateChildren(see TBaseVirtualTree.ValidateChildren Method , page 239)
Validates all children of a given node.

ValidateNode(see TBaseVirtualTree.ValidateNode Method , page 239)
Validates a given node.

ValidateNodeDataSize(see TBaseVirtualTree.ValidateNodeDataSize Method , page 239)
Helper method for node data size initalization.

WndProc(see TBaseVirtualTree.WndProc Method , page 240)
Redirected window procedure to do some special processing.

WriteChunks(see TBaseVirtualTree.WriteChunks Method , page 240)
Writes the core chunks for the given node to the given stream.

WriteNode(see TBaseVirtualTree.WriteNode Method , page 241)
Writes the cover (envelop) chunk for the given node to the given stream.

Properties

TBaseVirtualTree Class

Alignment(see TBaseVirtualTree.Alignment Property, page 107)
Determines the horizontal alignment of text if no columns are defined.

AnimationDuration(see TBaseVirtualTree.AnimationDuration Property, page 107)
Determines the maximum duration the tree can use to play an animation.

AutoExpandDelay(see TBaseVirtualTree.AutoExpandDelay Property, page 107)

Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay(see TBaseVirtualTree.AutoScrollDelay Property, page 108)
Time which determines when auto scrolling should start.

AutoScrollInterval(see TBaseVirtualTree.AutoScrollInterval Property, page 108)
Time interval between scroll events when doing auto scroll.

Background(see TBaseVirtualTree.Background Property, page 108)
Holds a background image for the tree.

BackgroundOffsetX(see TBaseVirtualTree.BackgroundOffsetX Property, page 109)
Horizontal offset of the background image.

BackgroundOffsetY(see TBaseVirtualTree.BackgroundOffsetY Property, page 109)
Vertical offset of the background image.

BorderStyle(see TBaseVirtualTree.BorderStyle Property, page 109)
Same as TForm.BorderStyle.

ButtonFillMode(see TBaseVirtualTree.ButtonFillMode Property, page 109)
Determines how to fill the background of the node buttons.

ButtonStyle(see TBaseVirtualTree.ButtonStyle Property, page 110)
Determines the look of node buttons.

ChangeDelay(see TBaseVirtualTree.ChangeDelay Property, page 110)
Time which determines when the OnChange(see TBaseVirtualTree.OnChange Event, page 131) event should be triggered after the actual change event.

CheckImageKind(see TBaseVirtualTree.CheckImageKind Property, page 110)
Determines which images should be used for checkboxes and radio buttons.

CheckImages(see TBaseVirtualTree.CheckImages Property, page 111)
Not documented.

CheckState(see TBaseVirtualTree.CheckState Property, page 111)
Read or set the check state of a node.

CheckType(see TBaseVirtualTree.CheckType Property, page 111)
Read or set the check type of a node.

ChildCount(see TBaseVirtualTree.ChildCount Property, page 111)
Read or set the number of child nodes of a node.

ChildrenInitialized(see TBaseVirtualTree.ChildrenInitialized Property, page 112)
Read whether a node's child count has been initialized already.

ClipboardFormats(see TBaseVirtualTree.ClipboardFormats Property, page 112)
Special class to keep a list of clipboard format descriptions.

Colors(see TBaseVirtualTree.Colors Property, page 112)
A collection of colors used in the tree.

CustomCheckImages(see TBaseVirtualTree.CustomCheckImages Property, page 113)
Assign your own image list to get the check images you like most.

DefaultNodeHeight(see TBaseVirtualTree.DefaultNodeHeight Property, page 113)
Read or set the height new nodes get as initial value.

DefaultPasteMode(see TBaseVirtualTree.DefaultPasteMode Property, page 113)
Read or set the value, which determines where to add pasted nodes to.

DragHeight(see TBaseVirtualTree.DragHeight Property, page 114)
Read or set the vertical limit of the internal drag image.

DragImage(see TBaseVirtualTree.DragImage Property, page 114)
Holds the instance of the internal drag image.

DragImageKind(see TBaseVirtualTree.DragImageKind Property, page 114)
Read or set what should be shown in the drag image.

DragManager(see TBaseVirtualTree.DragManager Property, page 114)
Holds the reference to the internal drag manager.

DragOperations(see TBaseVirtualTree.DragOperations Property, page 115)
Read or set which drag operations may be allowed in the tree.

DragSelection(see TBaseVirtualTree.DragSelection Property, page 115)
Keeps a temporary list of nodes during drag'n drop.

DragType(see TBaseVirtualTree.DragType Property, page 115)
Read or set which subsystem should be used for dragging(see TBaseVirtualTree.Dragging Method , page 198).

DragWidth(see TBaseVirtualTree.DragWidth Property, page 116)
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode(see TBaseVirtualTree.DrawSelectionMode Property, page 116)
Read or set how multiselection with the mouse is to be visualized.

DropTargetNode(see TBaseVirtualTree.DropTargetNode Property, page 116)
Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

EditColumn(see TBaseVirtualTree.EditColumn Property, page 117)
Not documented.

EditDelay(see TBaseVirtualTree.EditDelay Property, page 117)
Read or set the maximum time between two single clicks on the same node, which should start node editing.

EditLink(see TBaseVirtualTree.EditLink Property, page 117)
Keeps a reference to the internal edit link during a node edit operation.

Expanded(see TBaseVirtualTree.Expanded Property, page 118)
Read or set the expanded state of a particular node.

FocusedColumn(see TBaseVirtualTree.FocusedColumn Property, page 118)
Read or set the currently focused collumn.

FocusedNode(see TBaseVirtualTree.FocusedNode Property, page 118)
Read or set the currently focused node.

Font(see TBaseVirtualTree.Font Property, page 119)
Same as TWinControl.Font.

FullyVisible(see TBaseVirtualTree.FullyVisible Property, page 119)
Read or set whether a node is fully visible or not.

HasChildren(see TBaseVirtualTree.HasChildren Property, page 119)
Read or set whether a node has got children.

Header(see TBaseVirtualTree.Header Property, page 120)
Provides access to the header instance.

HeaderRect(see TBaseVirtualTree.HeaderRect Property, page 120)
Returns the non-client-area rectangle used for the header.

HintAnimation(see TBaseVirtualTree.HintAnimation Property, page 120)
Read or set the current hint animation type.

HintMode(see TBaseVirtualTree.HintMode Property, page 121)
Read or set what type of hint you want for the tree view.

HotCursor(see TBaseVirtualTree.HotCursor Property, page 121)
Read or set which cursor should be used for hot nodes.

HotNode(see TBaseVirtualTree.HotNode Property, page 121)
Read, which node is currently the hot node.

Images(see TBaseVirtualTree.Images Property, page 122)
Read or set the tree's normal image list.

IncrementalSearch(see TBaseVirtualTree.IncrementalSearch Property, page 122)
Read or set the current incremental search mode.

IncrementalSearchDirection(see TBaseVirtualTree.IncrementalSearchDirection Property, page 122)
Read or set the direction to be used for incremental search.

IncrementalSearchStart(see TBaseVirtualTree.IncrementalSearchStart Property, page 123)
Read or set where to start incremental search.

IncrementalSearchTimeout(see TBaseVirtualTree.IncrementalSearchTimeout Property, page 123)
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Indent(see TBaseVirtualTree.Indent Property, page 123)

Read or set the indentation amount for node levels.

IsDisabled(see TBaseVirtualTree.IsDisabled Property, page 124)
Read or set the enabled state of the given node.

IsVisible(see TBaseVirtualTree.IsVisible Property, page 124)
Read or set the visibility state of the given node.

LastClickPos(see TBaseVirtualTree.LastClickPos Property, page 124)
Used for retained drag start and wheel mouse scrolling.

LastDropMode(see TBaseVirtualTree.LastDropMode Property, page 125)
Read how the last drop operation finished.

LineMode(see TBaseVirtualTree.LineMode Property, page 125)
Read or set the mode of the tree lines.

LineStyle(see TBaseVirtualTree.LineStyle Property, page 125)
Read or set the mode of the tree lines.

Margin(see TBaseVirtualTree.Margin Property, page 125)
Read or set the tree's node margin.

MultiLine(see TBaseVirtualTree.MultiLine Property, page 126)
Read or toggle the multiline feature for a given node.

NodeAlignment(see TBaseVirtualTree.NodeAlignment Property, page 126)
Read or set the node alignment value.

NodeDataSize(see TBaseVirtualTree.NodeDataSize Property, page 127)
Read or set the extra data size for each node.

NodeHeight(see TBaseVirtualTree.NodeHeight Property, page 127)
Read or set a node's height.

NodeParent(see TBaseVirtualTree.NodeParent Property, page 127)
Read or set a node's parent node.

OffsetX(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

OffsetXY(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

OffsetY(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

RootNode(see TBaseVirtualTree.RootNode Property, page 153)
Reference to the internal root node which is the anchor of the entire tree node hierarchy.

RootNodeCount(see TBaseVirtualTree.RootNodeCount Property, page 153)
Read or set the number of nodes on the top level.

ScrollBarOptions(see TBaseVirtualTree.ScrollBarOptions Property, page 154)
Reference to the scroll bar options class.

SearchBuffer(see TBaseVirtualTree.SearchBuffer Property, page 154)
Current input string for incremental search.

Selected(see TBaseVirtualTree.Selected Property, page 154)
Property to modify or determine the selection state of a node.

SelectedCount(see TBaseVirtualTree.SelectedCount Property, page 155)
Contains the number of selected nodes.

SelectionBlendFactor(see TBaseVirtualTree.SelectionBlendFactor Property, page 155)
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

SelectionCurveRadius(see TBaseVirtualTree.SelectionCurveRadius Property, page 155)
Read or set the current corner radius for node selection rectangles.

StateImages(see TBaseVirtualTree.StateImages Property, page 156)
Reference to the images list which is used for the state images.

TextMargin(see TBaseVirtualTree.TextMargin Property, page 156)
Read or set the distance of the node caption to its borders.

TopNode(see TBaseVirtualTree.TopNode Property, page 157)
The top node is the node which is currently at the top border of the client area.

TotalCount(see TBaseVirtualTree.TotalCount Property, page 157)
Returns the number of nodes in the tree.

TotalInternalDataSize(see TBaseVirtualTree.TotalInternalDataSize Property, page 157)
Keeps the currently accumulated data size for one node.

TreeOptions(see TBaseVirtualTree.TreeOptions Property, page 158)
Reference to the tree's options.

TreeStates(see TBaseVirtualTree.TreeStates Property, page 158)
Property which keeps a set of flags which indicate current operation and states of the tree.

UpdateCount(see TBaseVirtualTree.UpdateCount Property, page 158)
Not documented.

VerticalAlignment(see TBaseVirtualTree.VerticalAlignment Property, page 158)
Used to set a node's vertical button aligment with regard to the entire node rectangle.

VisibleCount(see TBaseVirtualTree.VisibleCount Property, page 159)
Number of currently visible nodes.

VisiblePath(see TBaseVirtualTree.VisiblePath Property, page 159)
Property to set or determine a node parent's expand states.

WantTabs(see TBaseVirtualTree.WantTabs Property, page 159)
Read or set whether the tree wants to process tabs on its own.

Legend


protected


Event


Method

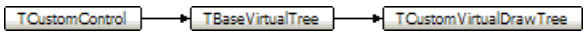

virtual


public


Property


read only

Class Hierarchy



File

VirtualTrees

10.1.8.1 TCustomVirtualDrawTree.OnDrawHint Event

Triggered when a node hint or tooltip must be drawn.

Pascal

```
property OnDrawHint: TVTDrawHintEvent;
```

Description

Use an event handler for OnDrawHint to draw the hint or tooltip for the given node. You must implement this event and OnGetHintSize(see TCustomVirtualDrawTree.OnGetHintSize Event, page 272) to get a hint at all.

Class

TCustomVirtualDrawTree Class(see page 252)

10.1.8.2 TCustomVirtualDrawTree.OnDrawNode Event

Triggered when a node must be drawn.

Pascal

```
property OnDrawNode: TVTDrawNodeEvent;
```

Description

Use an event handler for OnDrawNode to draw the actual content for the given node.

Class

TCustomVirtualDrawTree Class(see page 252)

10.1.8.3 TCustomVirtualDrawTree.OnGetHintSize Event

Triggered when a node hint or tooltip is about to show.

Pascal

```
property OnGetHintSize: TVTGetHintSizeEvent;
```

Description

Use an event handler for OnGetHintSize to return the size of the tooltip/hint window for the given node. You must implement this event and OnDrawHint(see TCustomVirtualDrawTree.OnDrawHint Event, page 271) to get a hint at all.

Class

TCustomVirtualDrawTree Class(see page 252)

10.1.8.4 TCustomVirtualDrawTree.OnGetNodeWidth Event

Triggered when a node is about to be drawn.

Pascal

```
property OnGetNodeWidth: TVTGetNodeWidthEvent;
```

Description

Use an event handler for OnGetNodeWidth to return your calculated width for the given node. Since the draw does not know the width of a node you have to tell it yourself.

Class

TCustomVirtualDrawTree Class(see page 252)

10.1.8.5 TCustomVirtualDrawTree.DoDrawHint Method

Overridable method which triggers OnDrawHint(see TCustomVirtualDrawTree.OnDrawHint Event, page 271).

Pascal

```
procedure DoDrawHint(Canvas: TCanvas; Node: PVirtualNode; R: TRect; Column: TColumnIndex);
```

Description

You can override DoDrawHint to customize the behavior for this request.

Class

TCustomVirtualDrawTree Class(see page 252)

10.1.8.6 TCustomVirtualDrawTree.DoGetHintSize Method

Overridable method which triggers OnGetHintSize(see TCustomVirtualDrawTree.OnGetHintSize Event, page 272).

Pascal

```
procedure DoGetHintSize(Node: PVirtualNode; Column: TColumnIndex; var R: TRect); virtual;
```

Description

You can override OnGetHintSize(see TCustomVirtualDrawTree.OnGetHintSize Event, page 272) to customize the behavior for this request.

Class

TCustomVirtualDrawTree Class(see page 252)

10.1.8.7 TCustomVirtualDrawTree.DoGetNodeWidth Method

Overridable method which triggers OnGetNodeWidth(see TCustomVirtualDrawTree.OnGetNodeWidth Event, page 272).

Pascal

```
function DoGetNodeWidth(Node: PVirtualNode; Column: TColumnIndex; Canvas: TCanvas = nil): Integer;
override;
```

Description

You can override OnGetNodeWidth(see TCustomVirtualDrawTree.OnGetNodeWidth Event, page 272) to customize the behavior for this request.

Class

TCustomVirtualDrawTree Class(see page 252)

10.1.8.8 TCustomVirtualDrawTree.DoPaintNode Method

Overridable method which triggers OnPaintNode.

Pascal

```
procedure DoPaintNode(var PaintInfo: TVTPaintInfo); override;
```

Description

You can override OnPaintNode to customize the behavior for this request.

Class

TCustomVirtualDrawTree Class(see page 252)

10.1.9 TCustomVirtualStringTree Class

Descendant of TBaseVirtualTree(see TBaseVirtualTree Class, page 88), which is able to manage node captions on its own

Pascal

```
TCustomVirtualStringTree = class(TBaseVirtualTree);
```

Description

TCustomVirtualStringTree enhances the base tree to display and edit node captions. It implements a generic node editor which can be used as reference to build your own one.

Events

OnGetHint(see TCustomVirtualStringTree.OnGetHint Event, page 295)
Virtual string tree event to query for a custom hint text.

OnGetText(see TCustomVirtualStringTree.OnGetText Event, page 295)
Virtual string tree event to query for a node's normal or static text.

OnNewText(see TCustomVirtualStringTree.OnNewText Event, page 296)
Virtual string tree event to pass edited text.

OnPaintText(see TCustomVirtualStringTree.OnPaintText Event, page 297)
Event to change text formatting for particular nodes.

OnShortenString(see TCustomVirtualStringTree.OnShortenString Event, page 297)
String tree event for custom handling of string abbreviations.

TBaseVirtualTree Class

OnAdvancedHeaderDraw(see TBaseVirtualTree.OnAdvancedHeaderDraw Event, page 128)
Header paint support event.

OnAfterCellPaint(see TBaseVirtualTree.OnAfterCellPaint Event, page 128)
Paint support event.

OnAfterItemErase(see TBaseVirtualTree.OnAfterItemErase Event, page 129)
Paint support event.

OnAfterItemPaint(see TBaseVirtualTree.OnAfterItemPaint Event, page 129)
Paint support event.

OnAfterPaint(see TBaseVirtualTree.OnAfterPaint Event, page 129)
Paint support event.

OnBeforeCellPaint(see TBaseVirtualTree.OnBeforeCellPaint Event, page 130)
Paint support event.

OnBeforeItemErase(see TBaseVirtualTree.OnBeforeItemErase Event, page 130)
Paint support event.

OnBeforeItemPaint(see TBaseVirtualTree.OnBeforeItemPaint Event, page 130)
Paint support event.

OnBeforePaint(see TBaseVirtualTree.OnBeforePaint Event, page 131)
Paint support event.

OnChange(see TBaseVirtualTree.OnChange Event, page 131)
Navigation support event.

OnChecked(see TBaseVirtualTree.OnChecked Event, page 131)
Check support event.

OnChecking(see TBaseVirtualTree.OnChecking Event, page 131)
Check support event.

OnCollapsed(see TBaseVirtualTree.OnCollapsed Event, page 132)

Miscellaneous event.

OnCollapsing(see TBaseVirtualTree.OnCollapsing Event, page 132)
Miscellaneous event.

OnColumnClick(see TBaseVirtualTree.OnColumnClick Event, page 132)
Header and column support event.

OnColumnDblClick(see TBaseVirtualTree.OnColumnDblClick Event, page 132)
Header and column support event.

OnColumnResize(see TBaseVirtualTree.OnColumnResize Event, page 133)
Header and column support routine.

OnCompareNodes(see TBaseVirtualTree.OnCompareNodes Event, page 133)
Sort and search support event.

OnCreateDataObject(see TBaseVirtualTree.OnCreateDataObject Event, page 134)
Drag'n drop support event.

OnCreateDragManager(see TBaseVirtualTree.OnCreateDragManager Event, page 134)
Drag'n drop support event.

OnCreateEditor(see TBaseVirtualTree.OnCreateEditor Event, page 134)
Editing support event.

OnDragAllowed(see TBaseVirtualTree.OnDragAllowed Event, page 135)
Drag'n drop support event.

OnDragDrop(see TBaseVirtualTree.OnDragDrop Event, page 135)
Drag'n drop support event.

OnDragOver(see TBaseVirtualTree.OnDragOver Event, page 137)
Drag'n drop support event.

OnEditCancelled(see TBaseVirtualTree.OnEditCancelled Event, page 137)
Editing support event.

OnEdited(see TBaseVirtualTree.OnEdited Event, page 137)
Editing support event.

OnEditing(see TBaseVirtualTree.OnEditing Event, page 138)
Editing support event.

OnExpanded(see TBaseVirtualTree.OnExpanded Event, page 138)
Misscellaneous event.

OnExpanding(see TBaseVirtualTree.OnExpanding Event, page 138)
Miscellaneous event.

OnFocusChanged(see TBaseVirtualTree.OnFocusChanged Event, page 138)
Navigation support event.

OnFocusChanging(see TBaseVirtualTree.OnFocusChanging Event, page 139)
Navigation support event.

OnFreeNode(see TBaseVirtualTree.OnFreeNode Event, page 139)
Data management node.

OnGetCellIsEmpty(see TBaseVirtualTree.OnGetCellIsEmpty Event, page 139)
Triggered when the tree control needs to know whether a given column is empty.

OnGetCursor(see TBaseVirtualTree.OnGetCursor Event, page 140)
Miscellaneous event.

OnGetHeaderCursor(see TBaseVirtualTree.OnGetHeaderCursor Event, page 140)
Header and column support event.

OnGetHelpContext(see TBaseVirtualTree.OnGetHelpContext Event, page 140)
Miscellaneous event.

OnGetImageIndex(see TBaseVirtualTree.OnGetImageIndex Event, page 140)
Display management event.

OnGetImageIndexEx(see TBaseVirtualTree.OnGetImageIndexEx Event, page 141)
Not documented.

OnGetLineStyle(see TBaseVirtualTree.OnGetLineStyle Event, page 141)
Display management event.

OnGetNodeDataSize(see TBaseVirtualTree.OnGetNodeDataSize Event, page 142)
Data management event.

OnGetPopupMenu(see TBaseVirtualTree.OnGetPopupMenu Event, page 142)
Miscellaneous event.

OnGetUserClipboardFormats(see TBaseVirtualTree.OnGetUserClipboardFormats Event, page 142)
Drag'n drop and clipboard support event.

OnHeaderClick(see TBaseVirtualTree.OnHeaderClick Event, page 143)
Header & column support event.

OnHeaderDblClick(see TBaseVirtualTree.OnHeaderDblClick Event, page 143)
Header & column support event.

OnHeaderDragged(see TBaseVirtualTree.OnHeaderDragged Event, page 143)
Header & column support event.

OnHeaderDraggedOut(see TBaseVirtualTree.OnHeaderDraggedOut Event, page 144)
Header & column support event.

OnHeaderDragging(see TBaseVirtualTree.OnHeaderDragging Event, page 144)
Header & column support event.

OnHeaderDraw(see TBaseVirtualTree.OnHeaderDraw Event, page 144)
Header & column support event.

OnHeaderDrawQueryElements(see TBaseVirtualTree.OnHeaderDrawQueryElements Event, page 145)
Header & column support event.

OnHeaderMouseDown(see TBaseVirtualTree.OnHeaderMouseDown Event, page 145)
Header & column support event.

OnHeaderMouseMove(see TBaseVirtualTree.OnHeaderMouseMove Event, page 145)
Header & column support event.

OnHeaderMouseUp(see TBaseVirtualTree.OnHeaderMouseUp Event, page 145)
Header & column support event.

OnHotChange(see TBaseVirtualTree.OnHotChange Event, page 146)
Navigation support event.

OnIncrementalSearch(see TBaseVirtualTree.OnIncrementalSearch Event, page 146)
Miscellaneous event.

OnInitChildren(see TBaseVirtualTree.OnInitChildren Event, page 147)
Node management event.

OnInitNode(see TBaseVirtualTree.OnInitNode Event, page 147)
Node management event.

OnKeyAction(see TBaseVirtualTree.OnKeyAction Event, page 148)
Miscellaneous event.

OnLoadNode(see TBaseVirtualTree.OnLoadNode Event, page 148)
Streaming support event.

OnMeasureItem(see TBaseVirtualTree.OnMeasureItem Event, page 148)
Miscellaneous event.

OnNodeCopied(see TBaseVirtualTree.OnNodeCopied Event, page 149)
Miscellaneous event.

OnNodeCopying(see TBaseVirtualTree.OnNodeCopying Event, page 149)
Miscellaneous event.

OnNodeMoved(see TBaseVirtualTree.OnNodeMoved Event, page 149)
Miscellaneous event.

OnNodeMoving(see TBaseVirtualTree.OnNodeMoving Event, page 150)
Miscellaneous event.

OnPaintBackground(see TBaseVirtualTree.OnPaintBackground Event, page 150)

Paint support event.

OnRenderOLEData(see TBaseVirtualTree.OnRenderOLEData Event, page 150)
Drag'n drop and clipboard support event.

OnResetNode(see TBaseVirtualTree.OnResetNode Event, page 151)
Node management event.

OnSaveNode(see TBaseVirtualTree.OnSaveNode Event, page 151)
Streaming support event.

OnScroll(see TBaseVirtualTree.OnScroll Event, page 152)
Miscellaneous event.

OnShowScrollbar(see TBaseVirtualTree.OnShowScrollbar Event, page 152)
Not documented.

OnStateChange(see TBaseVirtualTree.OnStateChange Event, page 152)
Miscellaneous event.

OnStructureChange(see TBaseVirtualTree.OnStructureChange Event, page 152)
Miscellaneous event.

OnUpdating(see TBaseVirtualTree.OnUpdating Event, page 153)
Miscellaneous event.

Group

Classes(see page 86)

Methods

AdjustPaintCellRect(see TCustomVirtualStringTree.AdjustPaintCellRect Method , page 298)
Method which can be used by descentants to adjust the given rectangle during a paint cycle.

CalculateTextWidth(see TCustomVirtualStringTree.CalculateTextWidth Method , page 299)
Not documented.

ColumnIsEmpty(see TCustomVirtualStringTree.ColumnIsEmpty Method , page 299)
Used to determine if a cell is considered as being empty.

ComputeNodeHeight(see TCustomVirtualStringTree.ComputeNodeHeight Method , page 299)
Not documented.

ContentToClipboard(see TCustomVirtualStringTree.ContentToClipboard Method , page 299)
Not documented.

ContentToHTML(see TCustomVirtualStringTree.ContentToClipboard Method , page 299)
Not documented.

ContentToRTF(see TCustomVirtualStringTree.ContentToClipboard Method , page 299)
Not documented.

ContentToText(see TCustomVirtualStringTree.ContentToClipboard Method , page 299)
Not documented.

ContentToUnicode(see TCustomVirtualStringTree.ContentToClipboard Method , page 299)
Not documented.

Create(see TCustomVirtualStringTree.Create Constructor , page 300)
Constructor of the control

DefineProperties(see TCustomVirtualStringTree.DefineProperties Method , page 300)
Helper method to customize loading and saving persistent tree data.

DoCreateEditor(see TCustomVirtualStringTree.DoCreateEditor Method , page 300)
Not documented.

DoGetNodeHint(see TCustomVirtualStringTree.DoGetNodeHint Method , page 301)
Not documented.

DoGetNodeTooltip(see TCustomVirtualStringTree.DoGetNodeTooltip Method , page 301)
Not documented.

DoGetNodeWidth(see TCustomVirtualStringTree.DoGetNodeWidth Method , page 301)
Overridable method which always retuns 0.

DoGetText(see TCustomVirtualStringTree.DoGetText Method , page 301)
Not documented.

DoIncrementalSearch(see TCustomVirtualStringTree.DoIncrementalSearch Method , page 302)
Not documented.

DoNewText(see TCustomVirtualStringTree.DoNewText Method , page 302)
Not documented.

DoPaintNode(see TCustomVirtualStringTree.DoPaintNode Method , page 302)
Overridable method which does nothing.

DoPaintText(see TCustomVirtualStringTree.DoPaintText Method , page 302)
Not documented.

DoShortenString(see TCustomVirtualStringTree.DoShortenString Method , page 303)
Not documented.

DoTextDrawing(see TCustomVirtualStringTree.DoTextDrawing Method , page 303)
Not documented.

DoTextMeasuring(see TCustomVirtualStringTree.DoTextMeasuring Method , page 303)
Not documented.

GetOptionsClass(see TCustomVirtualStringTree.GetOptionsClass Method , page 303)
Customization helper to determine which options class the tree should use.

GetTextInfo(see TCustomVirtualStringTree.GetTextInfo Method , page 304)
Helper method for node editors, hints etc.

InternalData(see TCustomVirtualStringTree.InternalData Method , page 305)
Returns the address of the internal data for a tree class.

InvalidateNode(see TCustomVirtualStringTree.InvalidateNode Method , page 305)
Invalidates the given node.

MainColumnChanged(see TCustomVirtualStringTree.MainColumnChanged Method , page 305)
Not documented.

Path(see TCustomVirtualStringTree.Path Method , page 306)
Not documented.

ReadChunk(see TCustomVirtualStringTree.ReadChunk Method , page 306)
Not documented.

ReadOldStringOptions(see TCustomVirtualStringTree.ReadOldStringOptions Method , page 306)
Not documented.

ReinitNode(see TCustomVirtualStringTree.ReinitNode Method , page 306)
Forces a reinitialization of the given node.

RenderOLEData(see TCustomVirtualStringTree.RenderOLEData Method , page 307)
Renders pending OLE data.

WriteChunks(see TCustomVirtualStringTree.WriteChunks Method , page 307)
Writes the core chunks for the given node to the given stream.

TBaseVirtualTree Class

AbsoluteIndex(see TBaseVirtualTree.AbsoluteIndex Method , page 160)
Reads the overall index of a node.

AddChild(see TBaseVirtualTree.AddChild Method , page 160)
Creates and adds a new child node to given node.

AddFromStream(see TBaseVirtualTree.AddFromStream Method , page 161)
Adds the content from the given stream to the given node.

AddToSelection(see TBaseVirtualTree.AddToSelection Method (PVirtualNode), page 161)
Adds one or more nodes to the current selection.

AdjustPaintCellRect(see TBaseVirtualTree.AdjustPaintCellRect Method , page 162)
Used in descentants to modify the clip rectangle of the current column while painting a certain node.

AdjustPanningCursor(see TBaseVirtualTree.AdjustPanningCursor Method , page 162)
Loads the proper cursor which indicates into which direction scrolling is done.

AdviseChangeEvent(see TBaseVirtualTree.AdviseChangeEvent Method , page 162)
Used to register a delayed change event.

AllocateInternalDataArea(see TBaseVirtualTree.AllocateInternalDataArea Method , page 163)
Registration method to allocate tree internal data per node.

Animate(see TBaseVirtualTree.Animate Method , page 163)
Support method for animated actions in the tree view.

Assign(see TBaseVirtualTree.Assign Method , page 164)
Used to copy properties from another Virtual Treeview.

BeginDrag(see TBaseVirtualTree.BeginDrag Method , page 164)
Starts an OLE drag'n drop operation.

BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164)
Enters the tree into a special synchronized mode.

BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164)
Locks the tree view to perform several update operations.

CalculateSelectionRect(see TBaseVirtualTree.CalculateSelectionRect Method , page 165)
Support method for draw selection.

CanAutoScroll(see TBaseVirtualTree.CanAutoScroll Method , page 165)
Determines whether the tree can currently auto scroll its window.

CancelCutOrCopy(see TBaseVirtualTree.CancelCutOrCopy Method , page 165)
Canceles any pending cut or copy clipboard operation.

CancelEditNode(see TBaseVirtualTree.CancelEditNode Method , page 166)
Cancel the current edit operation, if there is any.

CanEdit(see TBaseVirtualTree.CanEdit Method , page 166)
Determines whether a node can be edited or not.

CanFocus(see TBaseVirtualTree.CanFocus Method , page 166)
Support method to determine whether the tree window can receive the input focus.

CanShowDragImage(see TBaseVirtualTree.CanShowDragImage Method , page 166)
Determines whether a drag image should be shown.

Change(see TBaseVirtualTree.Change Method , page 167)
Central method called when a node's selection state changes.

ChangeScale(see TBaseVirtualTree.ChangeScale Method , page 167)
Helper method called by the VCL when control resizing is due.

CheckParentCheckState(see TBaseVirtualTree.CheckParentCheckState Method , page 167)
Helper method for recursive check state changes.

Clear(see TBaseVirtualTree.Clear Method , page 168)
Clears the tree and removes all nodes.

ClearChecked(see TBaseVirtualTree.ClearChecked Method , page 168)
Not documented.

ClearSelection(see TBaseVirtualTree.ClearSelection Method , page 168)
Removes all nodes from the current selection.

ClearTempCache(see TBaseVirtualTree.ClearTempCache Method , page 168)
Helper method to clear(see TBaseVirtualTree.Clear Method , page 168) the internal temporary node cache.

ColumnIsEmpty(see TBaseVirtualTree.ColumnIsEmpty Method , page 169)
Used to determine if a cell is considered as being empty.

CopyTo(see TBaseVirtualTree.CopyTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 169)
Copies Source and all its child nodes to Target.

CopyToClipBoard(see TBaseVirtualTree.CopyToClipBoard Method , page 169)
Copies all currently selected nodes to the clipboard.

CountLevelDifference(see TBaseVirtualTree.CountLevelDifference Method , page 170)
Determines the level difference of two nodes.

CountVisibleChildren(see TBaseVirtualTree.CountVisibleChildren Method , page 170)

Determines the number of visible child nodes of the given node.

Create(see TBaseVirtualTree.Create Constructor , page 170)
Constructor of the control

CreateParams(see TBaseVirtualTree.CreateParams Method , page 171)
Prepares the creation of the controls window handle.

CreateWnd(see TBaseVirtualTree.CreateWnd Method , page 171)
Initializes data, which depends on the window handle.

CutToClipBoard(see TBaseVirtualTree.CutToClipBoard Method , page 171)
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties(see TBaseVirtualTree.DefineProperties Method , page 171)
Helper method to customize loading and saving persistent tree data.

DeleteChildren(see TBaseVirtualTree.DeleteChildren Method , page 172)
Removes all child nodes from the given node.

DeleteNode(see TBaseVirtualTree.DeleteNode Method , page 172)
Removes the given node from the tree.

DeleteSelectedNodes(see TBaseVirtualTree.DeleteSelectedNodes Method , page 172)
Removes all currently selected nodes form the tree.

Destroy(see TBaseVirtualTree.Destroy Destructor , page 173)
Destructor of the control.

DetermineHiddenChildrenFlag(see TBaseVirtualTree.DetermineHiddenChildrenFlag Method , page 173)
Determines whether all children of a given node are hidden.

DetermineHiddenChildrenFlagAllNodes(see TBaseVirtualTree.DetermineHiddenChildrenFlagAllNodes Method , page 173)
Determines whether all children of all nodes are hidden.

DetermineHitPositionLTR(see TBaseVirtualTree.DetermineHitPositionLTR Method , page 174)
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineHitPositionRTL(see TBaseVirtualTree.DetermineHitPositionLTR Method , page 174)
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineNextCheckState(see TBaseVirtualTree.DetermineNextCheckState Method , page 174)
Not documented.

DetermineScrollDirections(see TBaseVirtualTree.DetermineScrollDirections Method , page 174)
Not documented.

DoAdvancedHeaderDraw(see TBaseVirtualTree.DoAdvancedHeaderDraw Method , page 174)
Not documented.

DoAfterCellPaint(see TBaseVirtualTree.DoAfterCellPaint Method , page 175)
Not documented.

DoAfterItemErase(see TBaseVirtualTree.DoAfterItemErase Method , page 175)
Not documented.

DoAfterItemPaint(see TBaseVirtualTree.DoAfterItemPaint Method , page 175)
Not documented.

DoAfterPaint(see TBaseVirtualTree.DoAfterPaint Method , page 175)
Not documented.

DoAutoScroll(see TBaseVirtualTree.DoAutoScroll Method , page 176)
Enables or disables the auto scroll timer.

DoBeforeCellPaint(see TBaseVirtualTree.DoBeforeCellPaint Method , page 176)
Not documented.

DoBeforeDrag(see TBaseVirtualTree.DoBeforeDrag Method , page 176)
Not documented.

DoBeforeItemErase(see TBaseVirtualTree.DoBeforeItemErase Method , page 177)
Not documented.

DoBeforeItemPaint(see TBaseVirtualTree.DoBeforeItemPaint Method , page 177)
Not documented.

DoBeforePaint(see TBaseVirtualTree.DoBeforePaint Method , page 177)
Not documented.

DoCancelEdit(see TBaseVirtualTree.DoCancelEdit Method , page 177)
Called when the tree should stop editing without accepting changed values.

DoCanEdit(see TBaseVirtualTree.DoCanEdit Method , page 178)
Not documented.

DoChange(see TBaseVirtualTree.DoChange Method , page 178)
Not documented.

DoCheckClick(see TBaseVirtualTree.DoCheckClick Method , page 178)
Not documented.

DoChecked(see TBaseVirtualTree.DoChecked Method , page 178)
Not documented.

DoChecking(see TBaseVirtualTree.DoChecking Method , page 179)
Not documented.

DoCollapsed(see TBaseVirtualTree.DoCollapsed Method , page 179)
Not documented.

DoCollapsing(see TBaseVirtualTree.DoCollapsing Method , page 179)
Not documented.

DoColumnClick(see TBaseVirtualTree.DoColumnClick Method , page 179)
Not documented.

DoColumnDblClick(see TBaseVirtualTree.DoColumnDblClick Method , page 180)
Not documented.

DoColumnResize(see TBaseVirtualTree.DoColumnResize Method , page 180)
Not documented.

DoCompare(see TBaseVirtualTree.DoCompare Method , page 180)
Not documented.

DoCreateDataObject(see TBaseVirtualTree.DoCreateDataObject Method , page 180)
Not documented.

DoCreateDragManager(see TBaseVirtualTree.DoCreateDragManager Method , page 181)
Not documented.

DoCreateEditor(see TBaseVirtualTree.DoCreateEditor Method , page 181)
Not documented.

DoDragDrop(see TBaseVirtualTree.DoDragDrop Method , page 181)
Not documented.

DoDragExpand(see TBaseVirtualTree.DoDragExpand Method , page 181)
Not documented.

DoDragging(see TBaseVirtualTree.DoDragging Method , page 182)
Internal method which handles drag' drop.

DoDragOver(see TBaseVirtualTree.DoDragOver Method , page 182)
Not documented.

DoEdit(see TBaseVirtualTree.DoEdit Method , page 182)
Initiates editing of the currently set focused column and edit node.

DoEndDrag(see TBaseVirtualTree.DoEndDrag Method , page 182)
Not documented.

DoEndEdit(see TBaseVirtualTree.DoEndEdit Method , page 183)
Stops the current edit operation and takes over the new content.

DoExpanded(see TBaseVirtualTree.DoExpanded Method , page 183)
Not documented.

DoExpanding(see TBaseVirtualTree.DoExpanding Method , page 183)
Not documented.

DoFocusChange(see TBaseVirtualTree.DoFocusChange Method , page 184)

Not documented.

DoFocusChanging(see TBaseVirtualTree.DoFocusChanging Method , page 184)
Not documented.

DoFocusNode(see TBaseVirtualTree.DoFocusNode Method , page 184)
Internal method to set the focused node.

DoFreeNode(see TBaseVirtualTree.DoFreeNode Method , page 184)
Not documented.

DoGetAnimationType(see TBaseVirtualTree.DoGetAnimationType Method , page 185)
Determines the type of animation to be used.

DoGetCursor(see TBaseVirtualTree.DoGetCursor Method , page 185)
Not documented.

DoGetHeaderCursor(see TBaseVirtualTree.DoGetHeaderCursor Method , page 185)
Not documented.

DoGetImageIndex(see TBaseVirtualTree.DoGetImageIndex Method , page 186)
Not documented.

DoGetLineStyle(see TBaseVirtualTree.DoGetLineStyle Method , page 186)
Not documented.

DoGetNodeHint(see TBaseVirtualTree.DoGetNodeHint Method , page 186)
Not documented.

DoGetNodeTooltip(see TBaseVirtualTree.DoGetNodeTooltip Method , page 186)
Not documented.

DoGetNodeWidth(see TBaseVirtualTree.DoGetNodeWidth Method , page 187)
Overridable method which always retuns 0.

DoGetPopupMenu(see TBaseVirtualTree.DoGetPopupMenu Method , page 187)
Overridable method which triggers the OnGetPopup event.

DoGetUserClipboardFormats(see TBaseVirtualTree.DoGetUserClipboardFormats Method , page 187)
Not documented.

DoHeaderClick(see TBaseVirtualTree.DoHeaderClick Method , page 187)
Not documented.

DoHeaderDblClick(see TBaseVirtualTree.DoHeaderDblClick Method , page 188)
Not documented.

DoHeaderDragged(see TBaseVirtualTree.DoHeaderDragged Method , page 188)
Not documented.

DoHeaderDraggedOut(see TBaseVirtualTree.DoHeaderDraggedOut Method , page 188)
Not documented.

DoHeaderDragging(see TBaseVirtualTree.DoHeaderDragging Method , page 189)
Not documented.

DoHeaderDraw(see TBaseVirtualTree.DoHeaderDraw Method , page 189)
Not documented.

DoHeaderDrawQueryElements(see TBaseVirtualTree.DoHeaderDrawQueryElements Method , page 189)
Not documented.

DoHeaderMouseDown(see TBaseVirtualTree.DoHeaderMouseDown Method , page 189)
Not documented.

DoHeaderMouseMove(see TBaseVirtualTree.DoHeaderMouseMove Method , page 190)
Not documented.

DoHeaderMouseUp(see TBaseVirtualTree.DoHeaderMouseUp Method , page 190)
Not documented.

DoHotChange(see TBaseVirtualTree.DoHotChange Method , page 190)
Not documented.

DoIncrementalSearch(see TBaseVirtualTree.DoIncrementalSearch Method , page 190)
Not documented.

DoInitChildren(see TBaseVirtualTree.DoInitChildren Method , page 191)
Not documented.

DoInitNode(see TBaseVirtualTree.DoInitNode Method , page 191)
Not documented.

DoKeyAction(see TBaseVirtualTree.DoKeyAction Method , page 191)
Not documented.

DoLoadUserData(see TBaseVirtualTree.DoLoadUserData Method , page 191)
Not documented.

DoMeasureItem(see TBaseVirtualTree.DoMeasureItem Method , page 192)
Not documented.

DoNodeCopied(see TBaseVirtualTree.DoNodeCopied Method , page 192)
Not documented.

DoNodeCopying(see TBaseVirtualTree.DoNodeCopying Method , page 192)
Not documented.

DoNodeMoved(see TBaseVirtualTree.DoNodeMoved Method , page 192)
Not documented.

DoNodeMoving(see TBaseVirtualTree.DoNodeMoving Method , page 193)
Not documented.

DoPaintBackground(see TBaseVirtualTree.DoPaintBackground Method , page 193)
Not documented.

DoPaintDropMark(see TBaseVirtualTree.DoPaintDropMark Method , page 193)
Overridable method which draws the small line on top of a nodes image depending on the current drop state.

DoPaintNode(see TBaseVirtualTree.DoPaintNode Method , page 193)
Overridable method which does nothing.

DoPopupMenu(see TBaseVirtualTree.DoPopupMenu Method , page 194)
Overridable method which shows the popup menu for the given node.

DoRenderOLEData(see TBaseVirtualTree.DoRenderOLEData Method , page 194)
Not documented.

DoReset(see TBaseVirtualTree.DoReset Method , page 194)
Not documented.

DoSaveUserData(see TBaseVirtualTree.DoSaveUserData Method , page 194)
Not documented.

DoScroll(see TBaseVirtualTree.DoScroll Method , page 195)
Overridable method which triggers the OnScroll(see TBaseVirtualTree.OnScroll Event, page 152) event.

DoSetOffsetXY(see TBaseVirtualTree.DoSetOffsetXY Method , page 195)
Internal core routine to set the tree's scroll position.

DoShowScrollbar(see TBaseVirtualTree.DoShowScrollbar Method , page 195)
Not documented.

DoStartDrag(see TBaseVirtualTree.DoStartDrag Method , page 196)
Not documented.

DoStateChange(see TBaseVirtualTree.DoStateChange Method , page 196)
Not documented.

DoStructureChange(see TBaseVirtualTree.DoStructureChange Method , page 196)
Not documented.

DoTimerScroll(see TBaseVirtualTree.DoTimerScroll Method , page 196)
Callback method which is triggered whenever the scroll timer fires.

DoUpdating(see TBaseVirtualTree.DoUpdating Method , page 197)
Not documented.

DoValidateCache(see TBaseVirtualTree.DoValidateCache Method , page 197)
Not documented.

DragCanceled(see TBaseVirtualTree.DragCanceled Method , page 197)

Called by the VCL when a drag'n drop operation was canceled by the user.

DragDrop(see TBaseVirtualTree.DragDrop Method , page 197)
Helper method, which is used when a drag operation is finished.

DragEnter(see TBaseVirtualTree.DragEnter Method , page 198)
Not documented.

DragFinished(see TBaseVirtualTree.DragFinished Method , page 198)
Called when a drag operation is finished (accepted or cancelled).

Dragging(see TBaseVirtualTree.Dragging Method , page 198)
Returns true if a drag'n drop operation is in progress.

DragLeave(see TBaseVirtualTree.DragLeave Method , page 199)
Not documented.

DragOver(see TBaseVirtualTree.DragOver Method , page 199)
Not documented.

DrawDottedHLine(see TBaseVirtualTree.DrawDottedHLine Method , page 199)
Not documented.

DrawDottedVLine(see TBaseVirtualTree.DrawDottedVLine Method , page 199)
Not documented.

EditNode(see TBaseVirtualTree.EditNode Method , page 200)
Starts editing the given node if allowed to.

EndEditNode(see TBaseVirtualTree.EndEditNode Method , page 200)
Stops node editing if it was started before.

EndSynch(see TBaseVirtualTree.EndSynch Method , page 200)
Counterpart to BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164).

EndUpdate(see TBaseVirtualTree.EndUpdate Method , page 201)
Resets the update lock set by BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164).

ExecuteAction(see TBaseVirtualTree.ExecuteAction Method , page 201)
Not documented.

FindNodeInSelection(see TBaseVirtualTree.FindNodeInSelection Method , page 201)
Helper method to find the given node in the current selection.

FinishChunkHeader(see TBaseVirtualTree.FinishChunkHeader Method , page 201)
Not documented.

FinishCutOrCopy(see TBaseVirtualTree.FinishCutOrCopy Method , page 202)
Stops any pending cut or copy clipboard operation.

FlushClipboard(see TBaseVirtualTree.FlushClipboard Method , page 202)
Renders all pending clipboard data.

FontChanged(see TBaseVirtualTree.FontChanged Method , page 202)
Not documented.

FullCollapse(see TBaseVirtualTree.FullCollapse Method , page 203)
Collapses all nodes in the tree.

FullExpand(see TBaseVirtualTree.FullExpand Method , page 203)
Expands all nodes in the tree.

GetBorderDimensions(see TBaseVirtualTree.GetBorderDimensions Method , page 203)
Not documented.

GetCheckImage(see TBaseVirtualTree.GetCheckImage Method , page 203)
Not documented.

GetCheckImageListFor(see TBaseVirtualTree.GetCheckImageListFor Method , page 204)
Not documented.

GetColumnClass(see TBaseVirtualTree.GetColumnClass Method , page 204)
Returns the class to be used to manage columns in the tree.

GetControlsAlignment(see TBaseVirtualTree.GetControlsAlignment Method , page 204)
Not documented.

GetDisplayRect(see TBaseVirtualTree.GetDisplayRect Method , page 205)
Returns the visible region used by the given node in client coordinates.

GetFirst(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstChecked(see TBaseVirtualTree.GetFirstChecked Method , page 206)
Not documented.

GetFirstChild(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstCutCopy(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstInitialized(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstSelected(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisible(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleChild(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleChildNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetHeaderClass(see TBaseVirtualTree.GetHeaderClass Method , page 206)
Returns the header class to be used by the tree.

GetHintWindowClass(see TBaseVirtualTree.GetHintWindowClass Method , page 206)
Not documented.

GetHitTestInfoAt(see TBaseVirtualTree.GetHitTestInfoAt Method , page 206)
Returns information about the node at the given position.

GetImageIndex(see TBaseVirtualTree.GetImageIndex Method , page 207)
Not documented.

GetLast(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastChild(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastChildNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastInitialized(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisible(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleChild(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleChildNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetMaxColumnWidth(see TBaseVirtualTree.GetMaxColumnWidth Method , page 208)

Returns the width of the largest node in the given column.

GetMaxRightExtend(see TBaseVirtualTree.GetMaxRightExtend Method , page 208)
Determines the maximum with of the currently visible part of the tree.

GetNativeClipboardFormats(see TBaseVirtualTree.GetNativeClipboardFormats Method , page 208)
Used to let descendants and the application add their own supported clipboard formats.

GetNext(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextChecked(see TBaseVirtualTree.GetNextChecked Method , page 209)
Not documented.

GetNextCutCopy(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextInitialized(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextSelected(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextSibling(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisible(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleSibling(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleSiblingNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNodeAt(see TBaseVirtualTree.GetNodeAt Method (Integer, Integer), page 210)
Not documented.

GetNodeData(see TBaseVirtualTree.GetNodeData Method , page 210)
Returns the address of the user data area of the given node.

GetNodeLevel(see TBaseVirtualTree.GetNodeLevel Method , page 210)
Returns the indentation level of the given node.

GetOptionsClass(see TBaseVirtualTree.GetOptionsClass Method , page 211)
Customization helper to determine which options class the tree should use.

GetPrevious(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousInitialized(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousSibling(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisible(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleSibling(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleSiblingNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetSortedCutCopySet(see TBaseVirtualTree.GetSortedCutCopySet Method , page 212)
Returns a sorted list of nodes, which are marked for s cut or copy clipboard operation.

GetSortedSelection(see TBaseVirtualTree.GetSortedSelection Method , page 213)
Returns a sorted list of all currently selected nodes.

GetTextInfo(see TBaseVirtualTree.GetTextInfo Method , page 213)
Helper method for node editors, hints etc.

GetTreeFromDataObject(see TBaseVirtualTree.GetTreeFromDataObject Method , page 213)
OLE drag'n drop and clipboard support method.

GetTreeRect(see TBaseVirtualTree.GetTreeRect Method , page 214)
Returns the size of the virtual tree image.

GetVisibleParent(see TBaseVirtualTree.GetVisibleParent Method , page 214)
Returns the first (nearest) parent node, which is visible.

HandleHotTrack(see TBaseVirtualTree.HandleHotTrack Method , page 214)
Not documented.

HandleIncrementalSearch(see TBaseVirtualTree.HandleIncrementalSearch Method , page 214)
Not documented.

HandleMouseDblClick(see TBaseVirtualTree.HandleMouseDblClick Method , page 215)
Not documented.

HandleMouseDown(see TBaseVirtualTree.HandleMouseDown Method , page 215)
Not documented.

HandleMouseUp(see TBaseVirtualTree.HandleMouseUp Method , page 215)
Not documented.

HasAsParent(see TBaseVirtualTree.HasAsParent Method , page 215)
Determines if the given node has got another node as one of its parents.

HasImage(see TBaseVirtualTree.HasImage Method , page 216)
Not documented.

HasPopupMenu(see TBaseVirtualTree.HasPopupMenu Method , page 216)
Determines whether there is a pop up menu assigned to the tree.

InitChildren(see TBaseVirtualTree.InitChildren Method , page 216)
Not documented.

InitNode(see TBaseVirtualTree.InitNode Method , page 217)
Not documented.

InsertNode(see TBaseVirtualTree.InsertNode Method , page 217)
Inserts a new node and returns it to the caller.

InternalAddFromStream(see TBaseVirtualTree.InternalAddFromStream Method , page 217)
Not documented.

InternalAddToSelection(see TBaseVirtualTree.InternalAddToSelection Method (PVirtualNode, Boolean), page 218)
Not documented.

InternalCacheNode(see TBaseVirtualTree.InternalCacheNode Method , page 218)
Not documented.

InternalClearSelection(see TBaseVirtualTree.InternalClearSelection Method , page 218)
Not documented.

InternalConnectNode(see TBaseVirtualTree.InternalConnectNode Method , page 219)
Not documented.

InternalData(see TBaseVirtualTree.InternalData Method , page 219)
Returns the address of the internal data for a tree class.

InternalDisconnectNode(see TBaseVirtualTree.InternalDisconnectNode Method , page 219)
Not documented.

InternalRemoveFromSelection(see TBaseVirtualTree.InternalRemoveFromSelection Method , page 220)
Not documented.

InvalidateCache(see TBaseVirtualTree.InvalidateCache Method , page 220)

Empties the internal node cache and marks it as invalid.

InvalidateChildren(see TBaseVirtualTree.InvalidateChildren Method , page 220)
Invalidates all children of the given node.

InvalidateColumn(see TBaseVirtualTree.InvalidateColumn Method , page 220)
Invalidates the client area part of a column.

InvalidateNode(see TBaseVirtualTree.InvalidateNode Method , page 221)
Invalidates the given node.

InvalidateToBottom(see TBaseVirtualTree.InvalidateToBottom Method , page 221)
Invalidates the client area starting with the top position of the given node.

InvertSelection(see TBaseVirtualTree.InvertSelection Method , page 221)
Inverts the current selection.

IsEditing(see TBaseVirtualTree.IsEditing Method , page 222)
Tells the caller whether the tree is currently in edit mode.

IsMouseSelecting(see TBaseVirtualTree.IsMouseSelecting Method , page 222)
Tell the caller whether the tree is currently in draw selection mode.

IterateSubtree(see TBaseVirtualTree.IterateSubtree Method , page 222)
Iterator method to go through all nodes of a given sub tree.

Loaded(see TBaseVirtualTree.Loaded Method , page 223)
Not documented.

LoadFromFile(see TBaseVirtualTree.LoadFromFile Method , page 223)
Loads previously streamed out tree data back in again.

LoadFromStream(see TBaseVirtualTree.LoadFromFile Method , page 223)
Loads previously streamed out tree data back in again.

MainColumnChanged(see TBaseVirtualTree.MainColumnChanged Method , page 223)
Not documented.

MarkCutCopyNodes(see TBaseVirtualTree.MarkCutCopyNodes Method , page 223)
Not documented.

MeasureItemHeight(see TBaseVirtualTree.MeasureItemHeight Method , page 224)
Not documented.

MouseMove(see TBaseVirtualTree.MouseMove Method , page 224)
Not documented.

MoveTo(see TBaseVirtualTree.MoveTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 224)
Moves Source and all its child nodes to Target.

Notification(see TBaseVirtualTree.Notification Method , page 225)
Not documented.

OriginalWMNCPaint(see TBaseVirtualTree.OriginalWMNCPaint Method , page 225)
Not documented.

Paint(see TBaseVirtualTree.Paint Method , page 225)
TControl's Paint method used here to display the tree.

PaintCheckImage(see TBaseVirtualTree.PaintCheckImage Method , page 225)
Not documented.

PaintImage(see TBaseVirtualTree.PaintImage Method , page 226)
Not documented.

PaintNodeButton(see TBaseVirtualTree.PaintNodeButton Method , page 226)
Not documented.

PaintSelectionRectangle(see TBaseVirtualTree.PaintSelectionRectangle Method , page 226)
Not documented.

PaintTree(see TBaseVirtualTree.PaintTree Method , page 226)
Main paint routine for the tree image.

PaintTreeLines(see TBaseVirtualTree.PaintTreeLines Method , page 227)
Not documented.

PanningWindowProc(see TBaseVirtualTree.PanningWindowProc Method , page 227)
Not documented.

PasteFromClipboard(see TBaseVirtualTree.PasteFromClipboard Method , page 227)
Inserts the content of the clipboard into the tree.

PrepareDragImage(see TBaseVirtualTree.PrepareDragImage Method , page 228)
Not documented.

Print(see TBaseVirtualTree.Print Method , page 228)
Not documented.

ProcessDrop(see TBaseVirtualTree.ProcessDrop Method , page 228)
Helper method to ease OLE drag'n drop operations.

ProcessOLEData(see TBaseVirtualTree.ProcessOLEData Method , page 229)
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk(see TBaseVirtualTree.ReadChunk Method , page 229)
Not documented.

ReadNode(see TBaseVirtualTree.ReadNode Method , page 229)
Not documented.

RedirectFontChangeEvent(see TBaseVirtualTree.RedirectFontChangeEvent Method , page 230)
Not documented.

ReinitChildren(see TBaseVirtualTree.ReinitChildren Method , page 230)
Forces all child nodes of Node to be reinitialized.

ReinitNode(see TBaseVirtualTree.ReinitNode Method , page 230)
Forces a reinitialization of the given node.

RemoveFromSelection(see TBaseVirtualTree.RemoveFromSelection Method , page 230)
Removes the given node from the current selection.

RenderOLEData(see TBaseVirtualTree.RenderOLEData Method , page 231)
Renders pending OLE data.

RepaintNode(see TBaseVirtualTree.RepaintNode Method , page 231)
Causes the treeview to repaint the given node.

ResetNode(see TBaseVirtualTree.ResetNode Method , page 231)
Resets the given node to uninitialized.

ResetRangeAnchor(see TBaseVirtualTree.ResetRangeAnchor Method , page 232)
Not documented.

RestoreFontChangeEvent(see TBaseVirtualTree.RestoreFontChangeEvent Method , page 232)
Not documented.

SaveToFile(see TBaseVirtualTree.SaveToFile Method , page 232)
Saves the entire content of the tree into a file or stream.

SaveToStream(see TBaseVirtualTree.SaveToFile Method , page 232)
Saves the entire content of the tree into a file or stream.

ScrollIntoView(see TBaseVirtualTree.ScrollIntoView Method , page 232)
Scrolls the tree so that the given node comes in the client area.

SelectAll(see TBaseVirtualTree.SelectAll Method , page 233)
Selects all nodes in the tree.

SelectNodes(see TBaseVirtualTree.SelectNodes Method , page 233)
Selects a range of nodes.

SetBiDiMode(see TBaseVirtualTree.SetBiDiMode Method , page 233)
Not documented.

SetFocusedNodeAndColumn(see TBaseVirtualTree.SetFocusedNodeAndColumn Method , page 234)
Not documented.

SkipNode(see TBaseVirtualTree.SkipNode Method , page 234)
Not documented.

Sort(see TBaseVirtualTree.Sort Method , page 234)

Sorts the given node.

SortTree(see TBaseVirtualTree.SortTree Method , page 234)
Sorts the entire tree view.

StartWheelPanning(see TBaseVirtualTree.StartWheelPanning Method , page 235)
Not documented.

StopWheelPanning(see TBaseVirtualTree.StopWheelPanning Method , page 235)
Not documented.

StructureChange(see TBaseVirtualTree.StructureChange Method , page 235)
Not documented.

SuggestDropEffect(see TBaseVirtualTree.SuggestDropEffect Method , page 236)
Not documented.

ToggleNode(see TBaseVirtualTree.ToggleNode Method , page 236)
Changes a node's expand state to the opposite state.

ToggleSelection(see TBaseVirtualTree.ToggleSelection Method , page 236)
Toggles the selection state of a range of nodes.

UnselectNodes(see TBaseVirtualTree.UnselectNodes Method , page 236)
Deselects a range of nodes.

UpdateAction(see TBaseVirtualTree.UpdateAction Method , page 237)
Not documented.

UpdateDesigner(see TBaseVirtualTree.UpdateDesigner Method , page 237)
Not documented.

UpdateEditBounds(see TBaseVirtualTree.UpdateEditBounds Method , page 237)
Not documented.

UpdateHeaderRect(see TBaseVirtualTree.UpdateHeaderRect Method , page 237)
Not documented.

UpdateHorizontalScrollBar(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateScrollBars(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateVerticalScrollBar(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateWindowAndDragImage(see TBaseVirtualTree.UpdateWindowAndDragImage Method , page 238)
Not documented.

UseRightToLeftReading(see TBaseVirtualTree.UseRightToLeftReading Method , page 238)
Helper method for right-to-left layout.

ValidateCache(see TBaseVirtualTree.ValidateCache Method , page 239)
Initiates the validation of the internal node cache.

ValidateChildren(see TBaseVirtualTree.ValidateChildren Method , page 239)
Validates all children of a given node.

ValidateNode(see TBaseVirtualTree.ValidateNode Method , page 239)
Validates a given node.

ValidateNodeDataSize(see TBaseVirtualTree.ValidateNodeDataSize Method , page 239)
Helper method for node data size initalization.

WndProc(see TBaseVirtualTree.WndProc Method , page 240)
Redirected window procedure to do some special processing.

WriteChunks(see TBaseVirtualTree.WriteChunks Method , page 240)
Writes the core chunks for the given node to the given stream.

WriteNode(see TBaseVirtualTree.WriteNode Method , page 241)
Writes the cover (envelop) chunk for the given node to the given stream.

Properties

DefaultText(see TCustomVirtualStringTree.DefaultText Property, page 295)

Not documented.

EllipsisWidth(see TCustomVirtualStringTree.EllipsisWidth Property, page 295)
Not documented.

Text(see TCustomVirtualStringTree.Text Property, page 298)
Not documented.

TreeOptions(see TCustomVirtualStringTree.TreeOptions Property, page 298)
Reference to the tree's options.

TBaseVirtualTree Class

Alignment(see TBaseVirtualTree.Alignment Property, page 107)
Determines the horizontal alignment of text if no columns are defined.

AnimationDuration(see TBaseVirtualTree.AnimationDuration Property, page 107)
Determines the maximum duration the tree can use to play an animation.

AutoExpandDelay(see TBaseVirtualTree.AutoExpandDelay Property, page 107)
Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay(see TBaseVirtualTree.AutoScrollDelay Property, page 108)
Time which determines when auto scrolling should start.

AutoScrollInterval(see TBaseVirtualTree.AutoScrollInterval Property, page 108)
Time interval between scroll events when doing auto scroll.

Background(see TBaseVirtualTree.Background Property, page 108)
Holds a background image for the tree.

BackgroundOffsetX(see TBaseVirtualTree.BackgroundOffsetX Property, page 109)
Horizontal offset of the background image.

BackgroundOffsetY(see TBaseVirtualTree.BackgroundOffsetY Property, page 109)
Vertical offset of the background image.

BorderStyle(see TBaseVirtualTree.BorderStyle Property, page 109)
Same as TForm.BorderStyle.

ButtonFillMode(see TBaseVirtualTree.ButtonFillMode Property, page 109)
Determines how to fill the background of the node buttons.

ButtonStyle(see TBaseVirtualTree.ButtonStyle Property, page 110)
Determines the look of node buttons.

ChangeDelay(see TBaseVirtualTree.ChangeDelay Property, page 110)
Time which determines when the OnChange(see TBaseVirtualTree.OnChange Event, page 131) event should be triggered after the actual change event.

CheckImageKind(see TBaseVirtualTree.CheckImageKind Property, page 110)
Determines which images should be used for checkboxes and radio buttons.

CheckImages(see TBaseVirtualTree.CheckImages Property, page 111)
Not documented.

CheckState(see TBaseVirtualTree.CheckState Property, page 111)
Read or set the check state of a node.

CheckType(see TBaseVirtualTree.CheckType Property, page 111)
Read or set the check type of a node.

ChildCount(see TBaseVirtualTree.ChildCount Property, page 111)
Read or set the number of child nodes of a node.

ChildrenInitialized(see TBaseVirtualTree.ChildrenInitialized Property, page 112)
Read whether a node's child count has been initialized already.

ClipboardFormats(see TBaseVirtualTree.ClipboardFormats Property, page 112)
Special class to keep a list of clipboard format descriptions.

Colors(see TBaseVirtualTree.Colors Property, page 112)
A collection of colors used in the tree.

CustomCheckImages(see TBaseVirtualTree.CustomCheckImages Property, page 113)
Assign your own image list to get the check images you like most.

DefaultNodeHeight(see TBaseVirtualTree.DefaultNodeHeight Property, page 113)
Read or set the height new nodes get as initial value.

DefaultPasteMode(see TBaseVirtualTree.DefaultPasteMode Property, page 113)
Read or set the value, which determines where to add pasted nodes to.

DragHeight(see TBaseVirtualTree.DragHeight Property, page 114)
Read or set the vertical limit of the internal drag image.

DragImage(see TBaseVirtualTree.DragImage Property, page 114)
Holds the instance of the internal drag image.

DragImageKind(see TBaseVirtualTree.DragImageKind Property, page 114)
Read or set what should be shown in the drag image.

DragManager(see TBaseVirtualTree.DragManager Property, page 114)
Holds the reference to the internal drag manager.

DragOperations(see TBaseVirtualTree.DragOperations Property, page 115)
Read or set which drag operations may be allowed in the tree.

DragSelection(see TBaseVirtualTree.DragSelection Property, page 115)
Keeps a temporary list of nodes during drag'n drop.

DragType(see TBaseVirtualTree.DragType Property, page 115)
Read or set which subsystem should be used for dragging(see TBaseVirtualTree.Dragging Method , page 198).

DragWidth(see TBaseVirtualTree.DragWidth Property, page 116)
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode(see TBaseVirtualTree.DrawSelectionMode Property, page 116)
Read or set how multiselection with the mouse is to be visualized.

DropTargetNode(see TBaseVirtualTree.DropTargetNode Property, page 116)
Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

EditColumn(see TBaseVirtualTree.EditColumn Property, page 117)
Not documented.

EditDelay(see TBaseVirtualTree.EditDelay Property, page 117)
Read or set the maximum time between two single clicks on the same node, which should start node editing.

EditLink(see TBaseVirtualTree.EditLink Property, page 117)
Keeps a reference to the internal edit link during a node edit operation.

Expanded(see TBaseVirtualTree.Expanded Property, page 118)
Read or set the expanded state of a particular node.

FocusedColumn(see TBaseVirtualTree.FocusedColumn Property, page 118)
Read or set the currently focused collumn.

FocusedNode(see TBaseVirtualTree.FocusedNode Property, page 118)
Read or set the currently focused node.

Font(see TBaseVirtualTree.Font Property, page 119)
Same as TWinControl.Font.

FullyVisible(see TBaseVirtualTree.FullyVisible Property, page 119)
Read or set whether a node is fully visible or not.

HasChildren(see TBaseVirtualTree.HasChildren Property, page 119)
Read or set whether a node has got children.

Header(see TBaseVirtualTree.Header Property, page 120)
Provides access to the header instance.

HeaderRect(see TBaseVirtualTree.HeaderRect Property, page 120)
Returns the non-client-area rectangle used for the header.

HintAnimation(see TBaseVirtualTree.HintAnimation Property, page 120)
Read or set the current hint animation type.

HintMode(see TBaseVirtualTree.HintMode Property, page 121)
Read or set what type of hint you want for the tree view.

HotCursor(see TBaseVirtualTree.HotCursor Property, page 121)

Read or set which cursor should be used for hot nodes.

HotNode(see TBaseVirtualTree.HotNode Property, page 121)
Read, which node is currently the hot node.

Images(see TBaseVirtualTree.Images Property, page 122)
Read or set the tree's normal image list.

IncrementalSearch(see TBaseVirtualTree.IncrementalSearch Property, page 122)
Read or set the current incremental search mode.

IncrementalSearchDirection(see TBaseVirtualTree.IncrementalSearchDirection Property, page 122)
Read or set the direction to be used for incremental search.

IncrementalSearchStart(see TBaseVirtualTree.IncrementalSearchStart Property, page 123)
Read or set where to start incremental search.

IncrementalSearchTimeout(see TBaseVirtualTree.IncrementalSearchTimeout Property, page 123)
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Indent(see TBaseVirtualTree.Indent Property, page 123)
Read or set the indentation amount for node levels.

IsDisabled(see TBaseVirtualTree.IsDisabled Property, page 124)
Read or set the enabled state of the given node.

IsVisible(see TBaseVirtualTree.IsVisible Property, page 124)
Read or set the visibility state of the given node.

LastClickPos(see TBaseVirtualTree.LastClickPos Property, page 124)
Used for retained drag start and wheel mouse scrolling.

LastDropMode(see TBaseVirtualTree.LastDropMode Property, page 125)
Read how the last drop operation finished.

LineMode(see TBaseVirtualTree.LineMode Property, page 125)
Read or set the mode of the tree lines.

LineStyle(see TBaseVirtualTree.LineStyle Property, page 125)
Read or set the mode of the tree lines.

Margin(see TBaseVirtualTree.Margin Property, page 125)
Read or set the tree's node margin.

MultiLine(see TBaseVirtualTree.MultiLine Property, page 126)
Read or toggle the multiline feature for a given node.

NodeAlignment(see TBaseVirtualTree.NodeAlignment Property, page 126)
Read or set the node alignment value.

NodeDataSize(see TBaseVirtualTree.NodeDataSize Property, page 127)
Read or set the extra data size for each node.

NodeHeight(see TBaseVirtualTree.NodeHeight Property, page 127)
Read or set a node's height.

NodeParent(see TBaseVirtualTree.NodeParent Property, page 127)
Read or set a node's parent node.

OffsetX(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

OffsetXY(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

OffsetY(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

RootNode(see TBaseVirtualTree.RootNode Property, page 153)
Reference to the internal root node which is the anchor of the entire tree node hierarchy.

RootNodeCount(see TBaseVirtualTree.RootNodeCount Property, page 153)
Read or set the number of nodes on the top level.

ScrollBarOptions(see TBaseVirtualTree.ScrollBarOptions Property, page 154)
Reference to the scroll bar options class.

SearchBuffer(see TBaseVirtualTree.SearchBuffer Property, page 154)
Current input string for incremental search.

Selected(see TBaseVirtualTree.Selected Property, page 154)
Property to modify or determine the selection state of a node.

SelectedCount(see TBaseVirtualTree.SelectedCount Property, page 155)
Contains the number of selected nodes.

SelectionBlendFactor(see TBaseVirtualTree.SelectionBlendFactor Property, page 155)
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

SelectionCurveRadius(see TBaseVirtualTree.SelectionCurveRadius Property, page 155)
Read or set the current corner radius for node selection rectangles.

StateImages(see TBaseVirtualTree.StateImages Property, page 156)
Reference to the images list which is used for the state images.

TextMargin(see TBaseVirtualTree.TextMargin Property, page 156)
Read or set the distance of the node caption to its borders.

TopNode(see TBaseVirtualTree.TopNode Property, page 157)
The top node is the node which is currently at the top border of the client area.

TotalCount(see TBaseVirtualTree.TotalCount Property, page 157)
Returns the number of nodes in the tree.

TotalInternalDataSize(see TBaseVirtualTree.TotalInternalDataSize Property, page 157)
Keeps the currently accumulated data size for one node.

TreeOptions(see TBaseVirtualTree.TreeOptions Property, page 158)
Reference to the tree's options.

TreeStates(see TBaseVirtualTree.TreeStates Property, page 158)
Property which keeps a set of flags which indicate current operation and states of the tree.

UpdateCount(see TBaseVirtualTree.UpdateCount Property, page 158)
Not documented.

VerticalAlignment(see TBaseVirtualTree.VerticalAlignment Property, page 158)
Used to set a node's vertical button aligment with regard to the entire node rectangle.

VisibleCount(see TBaseVirtualTree.VisibleCount Property, page 159)
Number of currently visible nodes.

VisiblePath(see TBaseVirtualTree.VisiblePath Property, page 159)
Property to set or determine a node parent's expand states.

WantTabs(see TBaseVirtualTree.WantTabs Property, page 159)
Read or set whether the tree wants to process tabs on its own.

Legend

protected

Property

read only

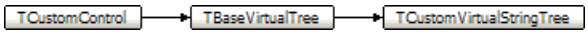
public

Event

Method

virtual

Class Hierarchy



File

VirtualTrees

10.1.9.1 TCustomVirtualStringTree.DefaultText Property

Not documented.

Pascal

```
property DefaultText: WideString;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.2 TCustomVirtualStringTree.EllipsisWidth Property

Not documented.

Pascal

```
property EllipsisWidth: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.3 TCustomVirtualStringTree.OnGetHint Event

Virtual string tree event to query for a custom hint text.

Pascal

```
property OnGetHint: TVSTGetHintEvent;
```

Description

Write an event handler for this event to specify a custom hint for the passed node and column. The TextType will always be ttNormal. This event will only be fired if HintMode(see TBaseVirtualTree.HintMode Property, page 121) is not hmTooltip. The delay for hints can be set as usual: adjust the properties HintPause and HintShortPause of the global Application object.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.4 TCustomVirtualStringTree.OnGetText Event

Virtual string tree event to query for a node's normal or static text.

Pascal

```
property OnGetText: TVSTGetTextEvent;
```

Description

This is one of the fundamental string tree events which must always be handled. The string tree will fire this event every time when it needs to know about the text of a specific node and column. This is mainly the case when the node appears in the visible area of the tree view (in other words it is not scrolled out of view) but also on some other occasions, including streaming, drag and drop and calculating the width of the node.

The node text is distinguished between two text types:

- Normal text: If TextType is ttNormal return the main node caption for the specified column.
- Static text: All text that you return when TextType is ttStatic will be displayed right beside the normal text (or left to it if the column's BidiMode is not bdLeftToRight, i.e. the column has right-to-left layout). Static text is used only for informational purposes; it cannot be selected or dragged and if the column is not wide enough to show all text it will not be shortened with an ellipsis (...) as normal text. The string tree will only query for static text if the StringOptions (see TreeOptions(see TCustomVirtualStringTree.TreeOptions Property, page 298)) include toShowStaticText. This is off by default.

When this event is fired the text parameter will always be initialized with the value of property DefaultText(see TCustomVirtualStringTree.DefaultText Property, page 295). To handle the event get your node data and then extract the string for the appropriate column and TextType.

Notes

Be sure that your event handler only contains absolutely necessary code. This event will be fired very often - easily a few hundred times for medium sized trees with some columns defined when the tree is repainted completely.

For example it is far too slow to use Locate() on some Dataset, a database query result or table, and then get the text from some TField. This may only work with in-memory tables or a client dataset. When you initialize your node data do some caching and use these cached values to display the data.

See Also

OnPaintText(see TCustomVirtualStringTree.OnPaintText Event, page 297)

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.5 TCustomVirtualStringTree.OnNewText Event

Virtual string tree event to pass edited text.

Pascal

```
property OnNewText: TVSTNewTextEvent;
```

Description

A string tree will fire this event after a node has been edited successfully (not canceled with Escape). The event handler must store the new text in the node data.

This event will only be used for the default node caption editor. Other custom node editors may or may not use this event to pass their edited data to the application. Editing for the whole tree is only possible if the MiscOptions (see TreeOptions(see TCustomVirtualStringTree.TreeOptions Property, page 298)) include toEditable. If only certain columns or nodes should be editable write an event handler for OnEditing(see TBaseVirtualTree.OnEditing Event, page 138).

See Also

OnCreateEditor(see TBaseVirtualTree.OnCreateEditor Event, page 134), OnEdited(see TBaseVirtualTree.OnEdited Event, page 137)

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.6 TCustomVirtualStringTree.OnPaintText Event

Event to change text formatting for particular nodes.

Pascal

```
property OnPaintText: TVTPaintText;
```

Description

Write an event handler for this event to render nodes with different fonts, font sizes, styles or colors. According to the parameters each column of each node and even normal and static text can be painted in different ways.

Notes

The string tree view manages an internal width for each node's main column. This is done because computing this width is quite costly and the width is needed on several occasions. If you change the font which is used to paint a node's text, for example to bold face style, its width changes but the tree view does not know this - it still relies on its cached node width. This may result in cut off selection rectangles among others.

Hence if the width of a node changes after its initialization because it is now formatted differently than before force a recalculation of the node width by calling InvalidateNode(see TCustomVirtualStringTree.InvalidateNode Method , page 305) (when the conditions for the changed formatting are met - not in the event handler for OnPaintText).

See Also

Paint cycles and stages(see page 36)

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.7 TCustomVirtualStringTree.OnShortenString Event

String tree event for custom handling of string abbreviations.

Pascal

```
property OnShortenString: TVSTShortenStringEvent;
```

Description

If the text of a node does not fit into its cell (in grid mode) or is too wide for the width of the tree view it is being abbreviated with an ellipsis (...). By default the ellipsis is added to the end of the node text.

Occasionally you may want to shorten the node text at a different position, for example if the node text is a path(see TCustomVirtualStringTree.Path Method , page 306) string and not the last folder or filename should be cut off but rather some mid level folders if possible.

In the handler S must be processed (shortened) and returned in Result. If Done is set to true (default value is false) the tree view takes over the shortening. This is useful if not all nodes or columns need

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.8 TCustomVirtualStringTree.Text Property

Not documented.

Pascal

```
property Text [Node: PVirtualNode; Column: TColumnIndex]: WideString;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.9 TCustomVirtualStringTree.TreeOptions Property

Reference to the tree's options.

Pascal

```
property TreeOptions: TCustomStringTreeOptions;
```

Description

The tree options are one of the main switchs to modify a treeview's behavior. Virtual Treeview supports customizing tree options by descentants. This allows very fine adjustments for derived tree classes, including the decision which properties should be published. For more information about the base options see TCustomVirtualTreeOptions and its descentants.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.10 TCustomVirtualStringTree.AdjustPaintCellRect Method

Method which can be used by descentants to adjust the given rectangle during a paint cycle.

Pascal

```
procedure AdjustPaintCellRect(var PaintInfo: TVTPaintInfo; var NextNonEmpty: TColumnIndex); override;
```

Description

For some special behaviour, like the auto span column feature, it is necessary to tell the base treeview which rectangle is to be considered as the current paint cell when drawing the tree. ClipRect is set to a rectangle which corresponds to the current node and the current column in the paint cycle.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.11 TCustomVirtualStringTree.CalculateTextWidth Method

Not documented.

Pascal

```
function CalculateTextWidth(Canvas: TCanvas; Node: PVirtualNode; Column: TColumnIndex; Text:
WideString): Integer; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.12 TCustomVirtualStringTree.ColumnIsEmpty Method

Used to determine if a cell is considered as being empty.

Pascal

```
function ColumnIsEmpty(Node: PVirtualNode; Column: TColumnIndex): Boolean; override;
```

Description

An empty cell might be used for the automatic column spanning feature. Descentants can override this method to modify the tree's behavior.

See Also

toAutoSpanColumns

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.13 TCustomVirtualStringTree.ComputeNodeHeight Method

Not documented.

Pascal

```
function ComputeNodeHeight(Canvas: TCanvas; Node: PVirtualNode; Column: TColumnIndex; S: WideString =
''): Integer; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.14 TCustomVirtualStringTree.ContentToClipboard Method

Not documented.

Pascal

```
function ContentToClipboard(Format: Word; Source: TVSTTextSourceType): HGLOBAL;
function ContentToHTML(Source: TVSTTextSourceType; Caption: WideString = ''): string;
```

```
function ContentToRTF(Source: TVSTTextSourceType): string;
function ContentToText(Source: TVSTTextSourceType; Separator: Char): string;
function ContentToUnicode(Source: TVSTTextSourceType; Separator: WideChar): WideString;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.15 TCustomVirtualStringTree.Create Constructor

Constructor of the control

Pascal

```
constructor Create(AOwner: TComponent); override;
```

Description

The constructor initializes certain properties to their default values.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.16 TCustomVirtualStringTree.DefineProperties Method

Helper method to customize loading and saving persistent tree data.

Pascal

```
procedure DefineProperties(Filer: TFiler); override;
```

Description

There were heavy changes in some properties during development of VT. This method helps to make migration easier by reading old properties manually and put them into the new properties as appropriate. These old properties are never written again and silently disappear.

Another task of this method is to work around the problem that TCollection is not streamed correctly when using Visual Form Inheritance (VFI).

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.17 TCustomVirtualStringTree.DoCreateEditor Method

Not documented.

Pascal

```
function DoCreateEditor(Node: PVirtualNode; Column: TColumnIndex): IVTEditLink; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.18 TCustomVirtualStringTree.DoGetNodeHint Method

Not documented.

Pascal

```
function DoGetNodeHint(Node: PVirtualNode; Column: TColumnIndex; var LineBreakStyle:
TVTTooltipLineBreakStyle): WideString; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.19 TCustomVirtualStringTree.DoGetNodeTooltip Method

Not documented.

Pascal

```
function DoGetNodeTooltip(Node: PVirtualNode; Column: TColumnIndex; var LineBreakStyle:
TVTTooltipLineBreakStyle): WideString; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.20 TCustomVirtualStringTree.DoGetNodeWidth Method

Overridable method which always retuns 0.

Pascal

```
function DoGetNodeWidth(Node: PVirtualNode; Column: TColumnIndex; Canvas: TCanvas = nil): Integer;
override;
```

Description

Descentants override this method to return a value which describes the width of a node. This is the inner width of the node excluding tree lines etc. So TVirtualStringTree returns the width of the node caption (plus text margin).

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.21 TCustomVirtualStringTree.DoGetText Method

Not documented.

Pascal

```
procedure DoGetText(Node: PVirtualNode; Column: TColumnIndex; TextType: TVSTTextType; var Text:
WideString); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.22 TCustomVirtualStringTree.DoIncrementalSearch Method

Not documented.

Pascal

```
function DoIncrementalSearch(Node: PVirtualNode; const Text: WideString): Integer; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.23 TCustomVirtualStringTree.DoNewText Method

Not documented.

Pascal

```
procedure DoNewText(Node: PVirtualNode; Column: TColumnIndex; Text: WideString); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.24 TCustomVirtualStringTree.DoPaintNode Method

Overridable method which does nothing.

Pascal

```
procedure DoPaintNode(var PaintInfo: TVTPaintInfo); override;
```

Description

Descentants override this method to paint the content of the node. For instance string trees draw the node's caption.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.25 TCustomVirtualStringTree.DoPaintText Method

Not documented.

Pascal

```
procedure DoPaintText(Node: PVirtualNode; const Canvas: TCanvas; Column: TColumnIndex; TextType:
```

```
TVSTTextType); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.26 TCustomVirtualStringTree.DoShortenString Method

Not documented.

Pascal

```
function DoShortenString(Canvas: TCanvas; Node: PVirtualNode; Column: TColumnIndex; const S:
WideString; Width: Integer; RightToLeft: Boolean; EllipsisWidth: Integer = 0): WideString; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.27 TCustomVirtualStringTree.DoTextDrawing Method

Not documented.

Pascal

```
procedure DoTextDrawing(var PaintInfo: TVTPaintInfo; Text: WideString; CellRect: TRect; DrawFormat:
Cardinal); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.28 TCustomVirtualStringTree.DoTextMeasuring Method

Not documented.

Pascal

```
function DoTextMeasuring(Canvas: TCanvas; Node: PVirtualNode; Column: TColumnIndex; Text: WideString):
Integer; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.29 TCustomVirtualStringTree.GetOptionsClass Method

Customization helper to determine which options class the tree should use.

Pascal

```
function GetOptionsClass: TTreeOptionsClass; override;
```

Description

GetOptionsClass is a special purpose method to return a certain class which is used by the tree for its options. TVirtualBaseTree always returns TCustomVirtualTreeOptions but descendants can override this method to return own classes.

For ease of use it makes much sense to always use the same name for the tree's options (which is TreeOptions). By using a customized options class, however, the wrong type is returned by this property. Hence it is meaningful to override TreeOptions and return the derived options class. To make this work the tree descendant must additionally provide new access methods for this property. An example can be seen in TVirtualStringTree:

```
  TVirtualStringTree = class(TCustomVirtualStringTree)
  private
    function GetOptions: TStringTreeOptions;
    procedure SetOptions(const Value: TStringTreeOptions);
  protected
    function GetOptionsClass: TTreeOptionsClass; override;
  public
    property Canvas;
  published
    ...
    property TreeOptions: TStringTreeOptions read GetOptions write SetOptions;
    ...
  end;

  ...

//----------------- TVirtualStringTree
--------------------------------------------------------------------------------

function TVirtualStringTree.GetOptions: TStringTreeOptions;

begin
  Result := FOptions as TStringTreeOptions;
end;

//----------------------------------------------------------------------------------------------------------
-----------------

procedure TVirtualStringTree.SetOptions(const Value: TStringTreeOptions);

begin
  FOptions.Assign(Value);
end;

//----------------------------------------------------------------------------------------------------------
-----------------

function TVirtualStringTree.GetOptionsClass: TTreeOptionsClass;

begin
  Result := TStringTreeOptions;
end;
```

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.30 TCustomVirtualStringTree.GetTextInfo Method

Helper method for node editors, hints etc.

Pascal

```
procedure GetTextInfo(Node: PVirtualNode; Column: TColumnIndex; const AFont: TFont; var R: TRect; var
Text: WideString); override;
```

Description

GetTextInfo is used to define a base access method for node data and the associated font from node editors and for hints.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.31 TCustomVirtualStringTree.InternalData Method

Returns the address of the internal data for a tree class.

Pascal

```
function InternalData(Node: PVirtualNode): Pointer;
```

Description

In TBaseVirtualTreeview this method returns nil but should be overridden in descendants to allow proper access to the internal data of Node if the descendant tree has allocated internal data.

See Also

Data handling(see page 39)

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.32 TCustomVirtualStringTree.InvalidateNode Method

Invalidates the given node.

Pascal

```
function InvalidateNode(Node: PVirtualNode): TRect; override;
```

Description

InvalidateNode initiates repaint of the given node by calling InvalidateRect with the node's display rectangel and returns this rectangle.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.33 TCustomVirtualStringTree.MainColumnChanged Method

Not documented.

Pascal

```
procedure MainColumnChanged; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.34 TCustomVirtualStringTree.Path Method

Not documented.

Pascal

```
function Path(Node: PVirtualNode; Column: TColumnIndex; TextType: TVSTTextType; Delimiter: WideChar):
WideString;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.35 TCustomVirtualStringTree.ReadChunk Method

Not documented.

Pascal

```
function ReadChunk(Stream: TStream; Version: Integer; Node: PVirtualNode; ChunkType: Integer;
ChunkSize: Integer): Boolean; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.36 TCustomVirtualStringTree.ReadOldStringOptions Method

Not documented.

Pascal

```
procedure ReadOldStringOptions(Reader: TReader);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.37 TCustomVirtualStringTree.ReinitNode Method

Forces a reinitialization of the given node.

Pascal

```
procedure ReinitNode(Node: PVirtualNode; Recursive: Boolean); override;
```

Description

ReinitNode forces Node and all its children (if Recursive is true) to be initialized again without modifying any data in the

nodes nor deleting children (unless the application requests a different amount).

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.38 TCustomVirtualStringTree.RenderOLEData Method

Renders pending OLE data.

Pascal

```
function RenderOLEData(const FormatEtcIn: TFormatEtc; out Medium: TStgMedium; ForClipboard: Boolean):
HResult; override;
```

Description

RenderOLData is called by TVTDataObject.GetData when a consumer of clipboard data actually requests the data. The base tree view only renders the native tree format, which is a chunk based stream of node data. The format to be rendered is specified in FormatEtcIn.cfFormat and is one of the formats which are returned from GetNativeClipboardFormats.

Descendants may override RenderOLEData in order to render other formats like HTML text. In TBaseVirtualTreeview this method calls the OnRenderOLEData event for all formats, except CF_VIRTUALTREE.

Class

TCustomVirtualStringTree Class(see page 274)

10.1.9.39 TCustomVirtualStringTree.WriteChunks Method

Writes the core chunks for the given node to the given stream.

Pascal

```
procedure WriteChunks(Stream: TStream; Node: PVirtualNode); override;
```

Description

WriteChunks is part of the streaming system in Virtual Treeview and writes the core chunks for Node into Stream. Descentants can optionally override this method to add other node specific chunks. This streaming is used when the tree must be saved to disk or a stream used e.g. for clipboard operations.

Notes

Keep in mind that this method is also called for the hidden root node. Using this fact in descendants you can create a

kind of "global" chunk set not directly bound to a specific node.

See Also

WriteNode, SaveToStream

Class

TCustomVirtualStringTree Class(see page 274)

10.1.10 TCustomVirtualTreeOptions Class

Organizes all tree options into subproperties for easier managment.

Pascal

```
TCustomVirtualTreeOptions = class(TPersistent);
```

Description

There are a lot of options available which control certain aspects of Virtual Treeview. Because there might only be at most 32 members in a published set and also for better overview these options have been splitted into several subsets, each related to a particular feature group like painting or node selection. With this implementation you can even derive an own option class and modify which options should be shown in Delphi's object inspector for your class.

Group

Classes(see page 86)

Methods

AssignTo(see TCustomVirtualTreeOptions.AssignTo Method , page 310)
Used to copy this option class to another option collection.

Create(see TCustomVirtualTreeOptions.Create Constructor , page 310)
Constructor of the class.

Properties

AnimationOptions(see TCustomVirtualTreeOptions.AnimationOptions Property, page 309)
Options related to animations.

AutoOptions(see TCustomVirtualTreeOptions.AutoOptions Property, page 309)
Options related to automatic actions.

MiscOptions(see TCustomVirtualTreeOptions.MiscOptions Property, page 309)
Options not related to any other category.

Owner(see TCustomVirtualTreeOptions.Owner Property, page 309)
Owner tree to which the property class belongs.

PaintOptions(see TCustomVirtualTreeOptions.PaintOptions Property, page 310)
Options related to painting.

SelectionOptions(see TCustomVirtualTreeOptions.SelectionOptions Property, page 310)
Options related to the way nodes can be selected.

Legend

protected

Property

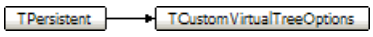
public

read only

Method

virtual

Class Hierarchy



File

VirtualTrees

10.1.10.1 TCustomVirtualTreeOptions.AnimationOptions Property

Options related to animations.

Pascal

```
property AnimationOptions: TVTAnimationOptions;
```

Description

These options can be used to switch certain animation effects in a tree.

Class

TCustomVirtualTreeOptions Class(see page 308)

10.1.10.2 TCustomVirtualTreeOptions.AutoOptions Property

Options related to automatic actions.

Pascal

```
property AutoOptions: TVTAutoOptions;
```

Description

These options can be used to switch certain actions in a tree which happen automatically under certain circumstances.

Class

TCustomVirtualTreeOptions Class(see page 308)

10.1.10.3 TCustomVirtualTreeOptions.MiscOptions Property

Options not related to any other category.

Pascal

```
property MiscOptions: TVTMiscOptions;
```

Description

These options can be used to switch miscellanous aspects in a tree.

Class

TCustomVirtualTreeOptions Class(see page 308)

10.1.10.4 TCustomVirtualTreeOptions.Owner Property

Owner tree to which the property class belongs.

Pascal

```
property Owner: TBaseVirtualTree;
```

Description

Owner tree to which the property class belongs.

Class

TCustomVirtualTreeOptions Class(see page 308)

10.1.10.5 TCustomVirtualTreeOptions.PaintOptions Property

Options related to painting.

Pascal

```
property PaintOptions: TVTPaintOptions;
```

Description

These options can be used to switch visual aspects of a tree.

Class

TCustomVirtualTreeOptions Class(see page 308)

10.1.10.6 TCustomVirtualTreeOptions.SelectionOptions Property

Options related to the way nodes can be selected.

Pascal

```
property SelectionOptions: TVTSelectionOptions;
```

Description

These options can be used to switch the way how nodes can be selected in a tree.

Class

TCustomVirtualTreeOptions Class(see page 308)

10.1.10.7 TCustomVirtualTreeOptions.AssignTo Method

Used to copy this option class to another option collection.

Pascal

```
procedure AssignTo(Dest: TPersistent); override;
```

Description

This is the usual method to support streaming or simply copying of this class. To stay open for future enhancements in form of new descentants not Assign but AssignTo has been used. AssignTo is called by TPersistent if there is no Assign method.

Class

TCustomVirtualTreeOptions Class(see page 308)

10.1.10.8 TCustomVirtualTreeOptions.Create Constructor

Constructor of the class.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree); virtual;
```

Description

Used to assign default values to all sub lists.

Class

TCustomVirtualTreeOptions Class(see page 308)

10.1.11 TEnumFormatEtc Class

```
TEnumFormatEtc = class(TInterfacedObject, IEnumFormatEtc);
```

Group

Classes(see page 86)

Methods

Clone(see TEnumFormatEtc.Clone Method , page 311)
Not documented.

Create(see TEnumFormatEtc.Create Constructor , page 312)
Not documented.

Next(see TEnumFormatEtc.Next Method , page 312)
Not documented.

Reset(see TEnumFormatEtc.Reset Method , page 312)
Not documented.

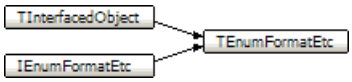
Skip(see TEnumFormatEtc.Skip Method , page 312)
Not documented.

Legend

public

Method

Class Hierarchy



File

VirtualTrees

10.1.11.1 TEnumFormatEtc.Clone Method

Not documented.

Pascal

```
function Clone(out Enum: IEnumFormatEtc): HResult; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TEnumFormatEtc Class(see page 311)

10.1.11.2 TEnumFormatEtc.Create Constructor

Not documented.

Pascal

```
constructor Create(Tree: TBaseVirtualTree; AFormatEtcArray: TFormatEtcArray);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TEnumFormatEtc Class(see page 311)

10.1.11.3 TEnumFormatEtc.Next Method

Not documented.

Pascal

```
function Next(celt: Integer; out elt; pceltFetched: PLongint): HResult; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TEnumFormatEtc Class(see page 311)

10.1.11.4 TEnumFormatEtc.Reset Method

Not documented.

Pascal

```
function Reset: HResult; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TEnumFormatEtc Class(see page 311)

10.1.11.5 TEnumFormatEtc.Skip Method

Not documented.

Pascal

```
function Skip(celt: Integer): HResult; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TEnumFormatEtc Class(see page 311)

10.1.12 TScrollBarOptions Class

```
TScrollBarOptions = class(TPersistent);
```

Group

Classes(see page 86)

Methods

Assign(see TScrollBarOptions.Assign Method , page 315)
Not documented.

Create(see TScrollBarOptions.Create Constructor , page 315)
Not documented.

GetOwner(see TScrollBarOptions.GetOwner Method , page 315)
Not documented.

Properties

AlwaysVisible(see TScrollBarOptions.AlwaysVisible Property, page 313)
Not documented.

HorizontalIncrement(see TScrollBarOptions.HorizontalIncrement Property, page 314)
Not documented.

ScrollBars(see TScrollBarOptions.ScrollBars Property, page 314)
Not documented.

ScrollBarStyle(see TScrollBarOptions.ScrollBarStyle Property, page 314)
Not documented.

VerticalIncrement(see TScrollBarOptions.VerticalIncrement Property, page 314)
Not documented.

Legend

published
Property
public
Method
virtual
protected

Class Hierarchy

File

VirtualTrees

10.1.12.1 TScrollBarOptions.AlwaysVisible Property

Not documented.

Pascal

```
property AlwaysVisible: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TScrollBarOptions Class(see page 313)

10.1.12.2 TScrollBarOptions.HorizontalIncrement Property

Not documented.

Pascal

```
property HorizontalIncrement: TVTScrollIncrement;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TScrollBarOptions Class(see page 313)

10.1.12.3 TScrollBarOptions.ScrollBars Property

Not documented.

Pascal

```
property ScrollBars: TScrollStyle;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TScrollBarOptions Class(see page 313)

10.1.12.4 TScrollBarOptions.ScrollBarStyle Property

Not documented.

Pascal

```
property ScrollBarStyle: TScrollBarStyle;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TScrollBarOptions Class(see page 313)

10.1.12.5 TScrollBarOptions.VerticalIncrement Property

Not documented.

Pascal

```
property VerticalIncrement: TVTScrollIncrement;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TScrollBarOptions Class(see page 313)

10.1.12.6 TScrollBarOptions.Assign Method

Not documented.

Pascal

```
procedure Assign(Source: TPersistent); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TScrollBarOptions Class(see page 313)

10.1.12.7 TScrollBarOptions.Create Constructor

Not documented.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TScrollBarOptions Class(see page 313)

10.1.12.8 TScrollBarOptions.GetOwner Method

Not documented.

Pascal

```
function GetOwner: TPersistent; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TScrollBarOptions Class(see page 313)

10.1.13 TStringEditLink Class

TStringEditLink is the standard node editor of a TVirtualStringTree(see TVirtualStringTree Class, page 402).

Pascal

```
TStringEditLink = class(TInterfacedObject, IVTEditLink);
```

Description

TStringEditLink implements the interface IVTEditLink(see IVTEditLink Interface, page 707). This is a simple node editor which wraps a TEdit and is not Unicode aware. A virtual string tree will use this node editor if the event OnCreateEditor is not handled and a node must be edited. After the node's text has been edited the event OnNewText will be fired and the application should replace the old text with the new and edited text.

The node editor instance will automatically be destroyed via reference counting when it is not needed anymore. Never destroy(see TStringEditLink.Destroy Destructor , page 318) it explicitly - except when you know what you are doing.

Remarks

If you want to modify some aspects of how the node editor works, i.e. suppress some characters or initialize it with a different text but the node's text, you can inherit your own class from TStringEditLink and return an instance of it in the OnCreateEditor event.

Group

Classes(see page 86)

Methods

BeginEdit(see TStringEditLink.BeginEdit Method , page 317)
This function will be called by the virtual string tree when the editing starts.

CancelEdit(see TStringEditLink.CancelEdit Method , page 318)
This function will be called by the virtual string tree when the current editing is about to be cancelled.

Create(see TStringEditLink.Create Constructor , page 318)
Constructor of the class.

Destroy(see TStringEditLink.Destroy Destructor , page 318)
Destructor of the class.

EndEdit(see TStringEditLink.EndEdit Method , page 318)
This function will be called by the virtual string tree when the current editing is being finished.

GetBounds(see TStringEditLink.GetBounds Method , page 319)
The virtual string tree uses this function to get the current bounding rect of the node editor.

PrepareEdit(see TStringEditLink.PrepareEdit Method , page 319)
This function is called by a virtual string tree to initialize the node editor.

ProcessMessage(see TStringEditLink.ProcessMessage Method , page 319)
This function is used to forward messages being directed to the virtual string tree.

SetBounds(see TStringEditLink.SetBounds Method , page 320)
The virtual string tree calls this function to initialize the bounding rect of the node editor.

IVTEditLink Interface

BeginEdit(see IVTEditLink.BeginEdit Method , page 708)
This function will be called by the virtual tree when the editing starts.

CancelEdit(see IVTEditLink.CancelEdit Method , page 709)
This function will be called by the virtual tree when the current editing is about to be cancelled.

EndEdit(see IVTEditLink.EndEdit Method , page 709)
This function will be called by the virtual tree when the current editing is being finished.

GetBounds(see IVTEditLink.GetBounds Method , page 710)
The virtual tree can use this function to get the current bounding rect of the node editor.

PrepareEdit(see IVTEditLink.PrepareEdit Method , page 710)
This function is called by a virtual tree to initialize the node editor.

ProcessMessage(see IVTEditLink.ProcessMessage Method , page 710)
This function is used to forward messages being directed to the virtual tree.

SetBounds(see IVTEditLink.SetBounds Method , page 711)
The virtual tree calls this function to initialize the bounding rectangle of the node editor.

Properties

Edit(see TStringEditLink.Edit Property, page 317)
Not documented.

Legend

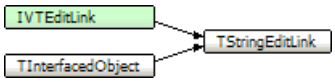
public

Property

Method

virtual

Class Hierarchy



File

VirtualTrees

10.1.13.1 TStringEditLink.Edit Property

Not documented.

Pascal

```
property Edit: TVTEdit;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TStringEditLink Class(see page 316)

10.1.13.2 TStringEditLink.BeginEdit Method

This function will be called by the virtual string tree when the editing starts.

Pascal

```
function BeginEdit: Boolean; virtual; stdcall;
```

Description

Please see interface IVTEditLink(see IVTEditLink Interface, page 707) for a detailed explanation of this interface

function.

Class

TStringEditLink Class(see page 316)

10.1.13.3 TStringEditLink.CancelEdit Method

This function will be called by the virtual string tree when the current editing is about to be cancelled.

Pascal

```
function CancelEdit: Boolean; virtual; stdcall;
```

Description

Please see interface IVTEditLink(see IVTEditLink Interface, page 707) for a detailed explanation of this interface function.

Class

TStringEditLink Class(see page 316)

10.1.13.4 TStringEditLink.Create Constructor

Constructor of the class.

Pascal

```
constructor Create;
```

Description

The constructor of the edit link also creates an instance of a simple node editor control. It is by default hidden and first displayed if the tree directs the link to do so.

Class

TStringEditLink Class(see page 316)

10.1.13.5 TStringEditLink.Destroy Destructor

Destructor of the class.

Pascal

```
destructor Destroy; override;
```

Description

Frees the internal editor control.

Class

TStringEditLink Class(see page 316)

10.1.13.6 TStringEditLink.EndEdit Method

This function will be called by the virtual string tree when the current editing is being finished.

Pascal

```
function EndEdit: Boolean; virtual; stdcall;
```

Description

Please see interface IVTEditLink(see IVTEditLink Interface, page 707) for a detailed explanation of this interface function.

Class

TStringEditLink Class(see page 316)

10.1.13.7 TStringEditLink.GetBounds Method

The virtual string tree uses this function to get the current bounding rect of the node editor.

Pascal

```
function GetBounds: TRect; virtual; stdcall;
```

Description

Please see interface IVTEditLink(see IVTEditLink Interface, page 707) for a detailed explanation of this interface function.

Class

TStringEditLink Class(see page 316)

10.1.13.8 TStringEditLink.PrepareEdit Method

This function is called by a virtual string tree to initialize the node editor.

Pascal

```
function PrepareEdit(Tree: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex): Boolean;
virtual; stdcall;
```

Description

Please see interface IVTEditLink(see IVTEditLink Interface, page 707) for a detailed explanation of this interface function.

Class

TStringEditLink Class(see page 316)

10.1.13.9 TStringEditLink.ProcessMessage Method

This function is used to forward messages being directed to the virtual string tree.

Pascal

```
procedure ProcessMessage(var Message: TMessage); virtual; stdcall;
```

Description

Please see interface IVTEditLink(see IVTEditLink Interface, page 707) for a detailed explanation of this interface function.

Class

TStringEditLink Class(see page 316)

10.1.13.10 TStringEditLink.SetBounds Method

The virtual string tree calls this function to initialize the bounding rect of the node editor.

Pascal

```
procedure SetBounds(R: TRect); virtual; stdcall;
```

Description

Please see interface IVTEditLink(see IVTEditLink Interface, page 707) for a detailed explanation of this interface function.

Class

TStringEditLink Class(see page 316)

10.1.14 TStringTreeOptions Class

Options class used in the string tree and its descentants.

Pascal

```
TStringTreeOptions = class(TCustomStringTreeOptions);
```

Description

This options class publishes all properties inherited from its ancestor and does not add any further functionality.

Group

Classes(see page 86)

Methods

TCustomStringTreeOptions Class

AssignTo(see TCustomStringTreeOptions.AssignTo Method , page 251)
Used to copy the options class.

Create(see TCustomStringTreeOptions.Create Constructor , page 251)
The constructor of the class.

TCustomVirtualTreeOptions Class

AssignTo(see TCustomVirtualTreeOptions.AssignTo Method , page 310)
Used to copy this option class to another option collection.

Create(see TCustomVirtualTreeOptions.Create Constructor , page 310)
Constructor of the class.

Properties

AnimationOptions(see TStringTreeOptions.AnimationOptions Property, page 321)
Options related to animations.

AutoOptions(see TStringTreeOptions.AutoOptions Property, page 322)
Options related to automatic actions.

MiscOptions(see TStringTreeOptions.MiscOptions Property, page 322)
Options not related to any other category.

PaintOptions(see TStringTreeOptions.PaintOptions Property, page 322)
Options related to painting.

SelectionOptions(see TStringTreeOptions.SelectionOptions Property, page 322)
Options related to the way nodes can be selected.

StringOptions(see TStringTreeOptions.StringOptions Property, page 323)
The new options introduced by the class.

TCustomStringTreeOptions Class

StringOptions(see TCustomStringTreeOptions.StringOptions Property, page 251)
The new options introduced by the class.

TCustomVirtualTreeOptions Class

AnimationOptions(see TCustomVirtualTreeOptions.AnimationOptions Property, page 309)
Options related to animations.

AutoOptions(see TCustomVirtualTreeOptions.AutoOptions Property, page 309)
Options related to automatic actions.

MiscOptions(see TCustomVirtualTreeOptions.MiscOptions Property, page 309)
Options not related to any other category.

Owner(see TCustomVirtualTreeOptions.Owner Property, page 309)
Owner tree to which the property class belongs.

PaintOptions(see TCustomVirtualTreeOptions.PaintOptions Property, page 310)
Options related to painting.

SelectionOptions(see TCustomVirtualTreeOptions.SelectionOptions Property, page 310)
Options related to the way nodes can be selected.

Legend

published

Property

protected

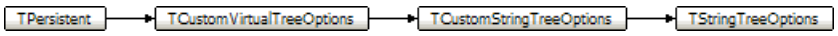
public

read only

Method

virtual

Class Hierarchy



File

VirtualTrees

10.1.14.1 TStringTreeOptions.AnimationOptions Property

Options related to animations.

Pascal

```
property AnimationOptions: TVTAnimationOptions;
```

Description

These options can be used to switch certain animation effects in a tree.

Class

TStringTreeOptions Class(see page 320)

10.1.14.2 TStringTreeOptions.AutoOptions Property

Options related to automatic actions.

Pascal

```
property AutoOptions: TVTAutoOptions;
```

Description

These options can be used to switch certain actions in a tree which happen automatically under certain circumstances.

Class

TStringTreeOptions Class(see page 320)

10.1.14.3 TStringTreeOptions.MiscOptions Property

Options not related to any other category.

Pascal

```
property MiscOptions: TVTMiscOptions;
```

Description

These options can be used to switch miscellanous aspects in a tree.

Class

TStringTreeOptions Class(see page 320)

10.1.14.4 TStringTreeOptions.PaintOptions Property

Options related to painting.

Pascal

```
property PaintOptions: TVTPaintOptions;
```

Description

These options can be used to switch visual aspects of a tree.

Class

TStringTreeOptions Class(see page 320)

10.1.14.5 TStringTreeOptions.SelectionOptions Property

Options related to the way nodes can be selected.

Pascal

```
property SelectionOptions: TVTSelectionOptions;
```

Description

These options can be used to switch the way how nodes can be selected in a tree.

Class

TStringTreeOptions Class(see page 320)

10.1.14.6 TStringTreeOptions.StringOptions Property

The new options introduced by the class.

Pascal

```
property StringOptions: TVTStringOptions;
```

Description

StringOptions provides access to the newly introduced options by which the base class is extended.

Class

TStringTreeOptions Class(see page 320)

10.1.15 TVirtualDrawTree Class

Descendant of TBaseVirtualTree(see TBaseVirtualTree Class, page 88), which passes node paint events through to the application (similar to a draw grid)

Pascal

```
TVirtualDrawTree = class(TCustomVirtualDrawTree);
```

Description

This tree implementation enhances the base tree to allow the application to draw its own stuff into the tree window.

Events

OnAdvancedHeaderDraw(see TVirtualDrawTree.OnAdvancedHeaderDraw Event, page 366)
Header paint support event.

OnAfterCellPaint(see TVirtualDrawTree.OnAfterCellPaint Event, page 367)
Paint support event.

OnAfterItemErase(see TVirtualDrawTree.OnAfterItemErase Event, page 367)
Paint support event.

OnAfterItemPaint(see TVirtualDrawTree.OnAfterItemPaint Event, page 367)
Paint support event.

OnAfterPaint(see TVirtualDrawTree.OnAfterPaint Event, page 368)
Paint support event.

OnBeforeCellPaint(see TVirtualDrawTree.OnBeforeCellPaint Event, page 368)
Paint support event.

OnBeforeItemErase(see TVirtualDrawTree.OnBeforeItemErase Event, page 368)
Paint support event.

OnBeforeItemPaint(see TVirtualDrawTree.OnBeforeItemPaint Event, page 369)
Paint support event.

OnBeforePaint(see TVirtualDrawTree.OnBeforePaint Event, page 369)
Paint support event.

OnChange(see TVirtualDrawTree.OnChange Event, page 369)
Navigation support event.

OnChecked(see TVirtualDrawTree.OnChecked Event, page 369)
Check support event.

OnChecking(see TVirtualDrawTree.OnChecking Event, page 370)
Check support event.

OnCollapsed(see TVirtualDrawTree.OnCollapsed Event, page 370)

Miscellaneous event.

OnCollapsing(see TVirtualDrawTree.OnCollapsing Event, page 370)
Miscellaneous event.

OnColumnClick(see TVirtualDrawTree.OnColumnClick Event, page 371)
Header and column support event.

OnColumnDblClick(see TVirtualDrawTree.OnColumnDblClick Event, page 371)
Header and column support event.

OnColumnResize(see TVirtualDrawTree.OnColumnResize Event, page 371)
Header and column support routine.

OnCompareNodes(see TVirtualDrawTree.OnCompareNodes Event, page 372)
Sort and search support event.

OnCreateDataObject(see TVirtualDrawTree.OnCreateDataObject Event, page 372)
Drag'n drop support event.

OnCreateDragManager(see TVirtualDrawTree.OnCreateDragManager Event, page 373)
Drag'n drop support event.

OnCreateEditor(see TVirtualDrawTree.OnCreateEditor Event, page 373)
Editing support event.

OnDragAllowed(see TVirtualDrawTree.OnDragAllowed Event, page 374)
Drag'n drop support event.

OnDragDrop(see TVirtualDrawTree.OnDragDrop Event, page 374)
Drag'n drop support event.

OnDragOver(see TVirtualDrawTree.OnDragOver Event, page 375)
Drag'n drop support event.

OnDrawHint(see TVirtualDrawTree.OnDrawHint Event, page 376)
Triggered when a node hint or tooltip must be drawn.

OnDrawNode(see TVirtualDrawTree.OnDrawNode Event, page 376)
Triggered when a node must be drawn.

OnEdited(see TVirtualDrawTree.OnEdited Event, page 376)
Editing support event.

OnEditing(see TVirtualDrawTree.OnEditing Event, page 377)
Editing support event.

OnExpanded(see TVirtualDrawTree.OnExpanded Event, page 378)
Misscellaneous event.

OnExpanding(see TVirtualDrawTree.OnExpanding Event, page 378)
Miscellaneous event.

OnFocusChanged(see TVirtualDrawTree.OnFocusChanged Event, page 378)
Navigation support event.

OnFocusChanging(see TVirtualDrawTree.OnFocusChanging Event, page 379)
Navigation support event.

OnFreeNode(see TVirtualDrawTree.OnFreeNode Event, page 379)
Data management node.

OnGetCellIsEmpty(see TVirtualDrawTree.OnGetCellIsEmpty Event, page 379)
Triggered when the tree control needs to know whether a given column is empty.

OnGetCursor(see TVirtualDrawTree.OnGetCursor Event, page 380)
Miscellaneous event.

OnGetHeaderCursor(see TVirtualDrawTree.OnGetHeaderCursor Event, page 380)
Header and column support event.

OnGetHelpContext(see TVirtualDrawTree.OnGetHelpContext Event, page 380)
Miscellaneous event.

OnGetHintSize(see TVirtualDrawTree.OnGetHintSize Event, page 380)
Triggered when a node hint or tooltip is about to show.

OnGetImageIndex(see TVirtualDrawTree.OnGetImageIndex Event, page 381)
Display management event.

OnGetImageIndexEx(see TVirtualDrawTree.OnGetImageIndexEx Event, page 381)
Not documented.

OnGetLineStyle(see TVirtualDrawTree.OnGetLineStyle Event, page 381)
Display management event.

OnGetNodeDataSize(see TVirtualDrawTree.OnGetNodeDataSize Event, page 382)
Data management event.

OnGetNodeWidth(see TVirtualDrawTree.OnGetNodeWidth Event, page 382)
Triggered when a node is about to be drawn.

OnGetPopupMenu(see TVirtualDrawTree.OnGetPopupMenu Event, page 383)
Miscellaneous event.

OnGetUserClipboardFormats(see TVirtualDrawTree.OnGetUserClipboardFormats Event, page 383)
Drag'n drop and clipboard support event.

OnHeaderClick(see TVirtualDrawTree.OnHeaderClick Event, page 383)
Header & column support event.

OnHeaderDblClick(see TVirtualDrawTree.OnHeaderDblClick Event, page 384)
Header & column support event.

OnHeaderDragged(see TVirtualDrawTree.OnHeaderDragged Event, page 384)
Header & column support event.

OnHeaderDraggedOut(see TVirtualDrawTree.OnHeaderDraggedOut Event, page 384)
Header & column support event.

OnHeaderDragging(see TVirtualDrawTree.OnHeaderDragging Event, page 385)
Header & column support event.

OnHeaderDraw(see TVirtualDrawTree.OnHeaderDraw Event, page 385)
Header & column support event.

OnHeaderDrawQueryElements(see TVirtualDrawTree.OnHeaderDrawQueryElements Event, page 385)
Header & column support event.

OnHeaderMouseDown(see TVirtualDrawTree.OnHeaderMouseDown Event, page 385)
Header & column support event.

OnHeaderMouseMove(see TVirtualDrawTree.OnHeaderMouseMove Event, page 386)
Header & column support event.

OnHeaderMouseUp(see TVirtualDrawTree.OnHeaderMouseUp Event, page 386)
Header & column support event.

OnHotChange(see TVirtualDrawTree.OnHotChange Event, page 386)
Navigation support event.

OnIncrementalSearch(see TVirtualDrawTree.OnIncrementalSearch Event, page 387)
Miscellaneous event.

OnInitChildren(see TVirtualDrawTree.OnInitChildren Event, page 387)
Node management event.

OnInitNode(see TVirtualDrawTree.OnInitNode Event, page 388)
Node management event.

OnKeyAction(see TVirtualDrawTree.OnKeyAction Event, page 388)
Miscellaneous event.

OnLoadNode(see TVirtualDrawTree.OnLoadNode Event, page 389)
Streaming support event.

OnMeasureItem(see TVirtualDrawTree.OnMeasureItem Event, page 390)
Miscellaneous event.

OnNodeCopied(see TVirtualDrawTree.OnNodeCopied Event, page 391)
Miscellaneous event.

OnNodeCopying(see TVirtualDrawTree.OnNodeCopying Event, page 391)

Miscellaneous event.

OnNodeMoved(see TVirtualDrawTree.OnNodeMoved Event, page 392)
Miscellaneous event.

OnNodeMoving(see TVirtualDrawTree.OnNodeMoving Event, page 392)
Miscellaneous event.

OnPaintBackground(see TVirtualDrawTree.OnPaintBackground Event, page 392)
Paint support event.

OnRenderOLEData(see TVirtualDrawTree.OnRenderOLEData Event, page 392)
Drag'n drop and clipboard support event.

OnResetNode(see TVirtualDrawTree.OnResetNode Event, page 393)
Node management event.

OnSaveNode(see TVirtualDrawTree.OnSaveNode Event, page 393)
Streaming support event.

OnScroll(see TVirtualDrawTree.OnScroll Event, page 394)
Miscellaneous event.

OnShowScrollbar(see TVirtualDrawTree.OnShowScrollbar Event, page 394)
Not documented.

OnStateChange(see TVirtualDrawTree.OnStateChange Event, page 395)
Miscellaneous event.

OnStructureChange(see TVirtualDrawTree.OnStructureChange Event, page 395)
Miscellaneous event.

OnUpdating(see TVirtualDrawTree.OnUpdating Event, page 395)
Miscellaneous event.

TCustomVirtualDrawTree Class

OnDrawHint(see TCustomVirtualDrawTree.OnDrawHint Event, page 271)
Triggered when a node hint or tooltip must be drawn.

OnDrawNode(see TCustomVirtualDrawTree.OnDrawNode Event, page 272)
Triggered when a node must be drawn.

OnGetHintSize(see TCustomVirtualDrawTree.OnGetHintSize Event, page 272)
Triggered when a node hint or tooltip is about to show.

OnGetNodeWidth(see TCustomVirtualDrawTree.OnGetNodeWidth Event, page 272)
Triggered when a node is about to be drawn.

TBaseVirtualTree Class

OnAdvancedHeaderDraw(see TBaseVirtualTree.OnAdvancedHeaderDraw Event, page 128)
Header paint support event.

OnAfterCellPaint(see TBaseVirtualTree.OnAfterCellPaint Event, page 128)
Paint support event.

OnAfterItemErase(see TBaseVirtualTree.OnAfterItemErase Event, page 129)
Paint support event.

OnAfterItemPaint(see TBaseVirtualTree.OnAfterItemPaint Event, page 129)
Paint support event.

OnAfterPaint(see TBaseVirtualTree.OnAfterPaint Event, page 129)
Paint support event.

OnBeforeCellPaint(see TBaseVirtualTree.OnBeforeCellPaint Event, page 130)
Paint support event.

OnBeforeItemErase(see TBaseVirtualTree.OnBeforeItemErase Event, page 130)
Paint support event.

OnBeforeItemPaint(see TBaseVirtualTree.OnBeforeItemPaint Event, page 130)
Paint support event.

OnBeforePaint(see TBaseVirtualTree.OnBeforePaint Event, page 131)

Paint support event.

OnChange(see TBaseVirtualTree.OnChange Event, page 131)
Navigation support event.

OnChecked(see TBaseVirtualTree.OnChecked Event, page 131)
Check support event.

OnChecking(see TBaseVirtualTree.OnChecking Event, page 131)
Check support event.

OnCollapsed(see TBaseVirtualTree.OnCollapsed Event, page 132)
Miscellaneous event.

OnCollapsing(see TBaseVirtualTree.OnCollapsing Event, page 132)
Miscellaneous event.

OnColumnClick(see TBaseVirtualTree.OnColumnClick Event, page 132)
Header and column support event.

OnColumnDblClick(see TBaseVirtualTree.OnColumnDblClick Event, page 132)
Header and column support event.

OnColumnResize(see TBaseVirtualTree.OnColumnResize Event, page 133)
Header and column support routine.

OnCompareNodes(see TBaseVirtualTree.OnCompareNodes Event, page 133)
Sort and search support event.

OnCreateDataObject(see TBaseVirtualTree.OnCreateDataObject Event, page 134)
Drag'n drop support event.

OnCreateDragManager(see TBaseVirtualTree.OnCreateDragManager Event, page 134)
Drag'n drop support event.

OnCreateEditor(see TBaseVirtualTree.OnCreateEditor Event, page 134)
Editing support event.

OnDragAllowed(see TBaseVirtualTree.OnDragAllowed Event, page 135)
Drag'n drop support event.

OnDragDrop(see TBaseVirtualTree.OnDragDrop Event, page 135)
Drag'n drop support event.

OnDragOver(see TBaseVirtualTree.OnDragOver Event, page 137)
Drag'n drop support event.

OnEditCancelled(see TBaseVirtualTree.OnEditCancelled Event, page 137)
Editing support event.

OnEdited(see TBaseVirtualTree.OnEdited Event, page 137)
Editing support event.

OnEditing(see TBaseVirtualTree.OnEditing Event, page 138)
Editing support event.

OnExpanded(see TBaseVirtualTree.OnExpanded Event, page 138)
Misscellaneous event.

OnExpanding(see TBaseVirtualTree.OnExpanding Event, page 138)
Miscellaneous event.

OnFocusChanged(see TBaseVirtualTree.OnFocusChanged Event, page 138)
Navigation support event.

OnFocusChanging(see TBaseVirtualTree.OnFocusChanging Event, page 139)
Navigation support event.

OnFreeNode(see TBaseVirtualTree.OnFreeNode Event, page 139)
Data management node.

OnGetCellIsEmpty(see TBaseVirtualTree.OnGetCellIsEmpty Event, page 139)
Triggered when the tree control needs to know whether a given column is empty.

OnGetCursor(see TBaseVirtualTree.OnGetCursor Event, page 140)
Miscellaneous event.

OnGetHeaderCursor(see TBaseVirtualTree.OnGetHeaderCursor Event, page 140)
Header and column support event.

OnGetHelpContext(see TBaseVirtualTree.OnGetHelpContext Event, page 140)
Miscellaneous event.

OnGetImageIndex(see TBaseVirtualTree.OnGetImageIndex Event, page 140)
Display management event.

OnGetImageIndexEx(see TBaseVirtualTree.OnGetImageIndexEx Event, page 141)
Not documented.

OnGetLineStyle(see TBaseVirtualTree.OnGetLineStyle Event, page 141)
Display management event.

OnGetNodeDataSize(see TBaseVirtualTree.OnGetNodeDataSize Event, page 142)
Data management event.

OnGetPopupMenu(see TBaseVirtualTree.OnGetPopupMenu Event, page 142)
Miscellaneous event.

OnGetUserClipboardFormats(see TBaseVirtualTree.OnGetUserClipboardFormats Event, page 142)
Drag'n drop and clipboard support event.

OnHeaderClick(see TBaseVirtualTree.OnHeaderClick Event, page 143)
Header & column support event.

OnHeaderDblClick(see TBaseVirtualTree.OnHeaderDblClick Event, page 143)
Header & column support event.

OnHeaderDragged(see TBaseVirtualTree.OnHeaderDragged Event, page 143)
Header & column support event.

OnHeaderDraggedOut(see TBaseVirtualTree.OnHeaderDraggedOut Event, page 144)
Header & column support event.

OnHeaderDragging(see TBaseVirtualTree.OnHeaderDragging Event, page 144)
Header & column support event.

OnHeaderDraw(see TBaseVirtualTree.OnHeaderDraw Event, page 144)
Header & column support event.

OnHeaderDrawQueryElements(see TBaseVirtualTree.OnHeaderDrawQueryElements Event, page 145)
Header & column support event.

OnHeaderMouseDown(see TBaseVirtualTree.OnHeaderMouseDown Event, page 145)
Header & column support event.

OnHeaderMouseMove(see TBaseVirtualTree.OnHeaderMouseMove Event, page 145)
Header & column support event.

OnHeaderMouseUp(see TBaseVirtualTree.OnHeaderMouseUp Event, page 145)
Header & column support event.

OnHotChange(see TBaseVirtualTree.OnHotChange Event, page 146)
Navigation support event.

OnIncrementalSearch(see TBaseVirtualTree.OnIncrementalSearch Event, page 146)
Miscellaneous event.

OnInitChildren(see TBaseVirtualTree.OnInitChildren Event, page 147)
Node management event.

OnInitNode(see TBaseVirtualTree.OnInitNode Event, page 147)
Node management event.

OnKeyAction(see TBaseVirtualTree.OnKeyAction Event, page 148)
Miscellaneous event.

OnLoadNode(see TBaseVirtualTree.OnLoadNode Event, page 148)
Streaming support event.

OnMeasureItem(see TBaseVirtualTree.OnMeasureItem Event, page 148)
Miscellaneous event.

OnNodeCopied(see TBaseVirtualTree.OnNodeCopied Event, page 149)

Miscellaneous event.

OnNodeCopying(see TBaseVirtualTree.OnNodeCopying Event, page 149)
Miscellaneous event.

OnNodeMoved(see TBaseVirtualTree.OnNodeMoved Event, page 149)
Miscellaneous event.

OnNodeMoving(see TBaseVirtualTree.OnNodeMoving Event, page 150)
Miscellaneous event.

OnPaintBackground(see TBaseVirtualTree.OnPaintBackground Event, page 150)
Paint support event.

OnRenderOLEData(see TBaseVirtualTree.OnRenderOLEData Event, page 150)
Drag'n drop and clipboard support event.

OnResetNode(see TBaseVirtualTree.OnResetNode Event, page 151)
Node management event.

OnSaveNode(see TBaseVirtualTree.OnSaveNode Event, page 151)
Streaming support event.

OnScroll(see TBaseVirtualTree.OnScroll Event, page 152)
Miscellaneous event.

OnShowScrollbar(see TBaseVirtualTree.OnShowScrollbar Event, page 152)
Not documented.

OnStateChange(see TBaseVirtualTree.OnStateChange Event, page 152)
Miscellaneous event.

OnStructureChange(see TBaseVirtualTree.OnStructureChange Event, page 152)
Miscellaneous event.

OnUpdating(see TBaseVirtualTree.OnUpdating Event, page 153)
Miscellaneous event.

Group

Classes(see page 86)

Methods

GetOptionsClass(see TVirtualDrawTree.GetOptionsClass Method , page 401)
Customization helper to determine which options class the tree should use.

TCustomVirtualDrawTree Class

DoDrawHint(see TCustomVirtualDrawTree.DoDrawHint Method , page 272)
Overridable method which triggers OnDrawHint(see TCustomVirtualDrawTree.OnDrawHint Event, page 271).

DoGetHintSize(see TCustomVirtualDrawTree.DoGetHintSize Method , page 273)
Overridable method which triggers OnGetHintSize(see TCustomVirtualDrawTree.OnGetHintSize Event, page 272).

DoGetNodeWidth(see TCustomVirtualDrawTree.DoGetNodeWidth Method , page 273)
Overridable method which triggers OnGetNodeWidth(see TCustomVirtualDrawTree.OnGetNodeWidth Event, page 272).

DoPaintNode(see TCustomVirtualDrawTree.DoPaintNode Method , page 273)
Overridable method which triggers OnPaintNode.

TBaseVirtualTree Class

AbsoluteIndex(see TBaseVirtualTree.AbsoluteIndex Method , page 160)
Reads the overall index of a node.

AddChild(see TBaseVirtualTree.AddChild Method , page 160)
Creates and adds a new child node to given node.

AddFromStream(see TBaseVirtualTree.AddFromStream Method , page 161)
Adds the content from the given stream to the given node.

AddToSelection(see TBaseVirtualTree.AddToSelection Method (PVirtualNode), page 161)
Adds one or more nodes to the current selection.

AdjustPaintCellRect(see TBaseVirtualTree.AdjustPaintCellRect Method , page 162)

Used in descentants to modify the clip rectangle of the current column while painting a certain node.

AdjustPanningCursor(see TBaseVirtualTree.AdjustPanningCursor Method , page 162)
Loads the proper cursor which indicates into which direction scrolling is done.

AdviseChangeEvent(see TBaseVirtualTree.AdviseChangeEvent Method , page 162)
Used to register a delayed change event.

AllocateInternalDataArea(see TBaseVirtualTree.AllocateInternalDataArea Method , page 163)
Registration method to allocate tree internal data per node.

Animate(see TBaseVirtualTree.Animate Method , page 163)
Support method for animated actions in the tree view.

Assign(see TBaseVirtualTree.Assign Method , page 164)
Used to copy properties from another Virtual Treeview.

BeginDrag(see TBaseVirtualTree.BeginDrag Method , page 164)
Starts an OLE drag'n drop operation.

BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164)
Enters the tree into a special synchronized mode.

BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164)
Locks the tree view to perform several update operations.

CalculateSelectionRect(see TBaseVirtualTree.CalculateSelectionRect Method , page 165)
Support method for draw selection.

CanAutoScroll(see TBaseVirtualTree.CanAutoScroll Method , page 165)
Determines whether the tree can currently auto scroll its window.

CancelCutOrCopy(see TBaseVirtualTree.CancelCutOrCopy Method , page 165)
Canceles any pending cut or copy clipboard operation.

CancelEditNode(see TBaseVirtualTree.CancelEditNode Method , page 166)
Cancel the current edit operation, if there is any.

CanEdit(see TBaseVirtualTree.CanEdit Method , page 166)
Determines whether a node can be edited or not.

CanFocus(see TBaseVirtualTree.CanFocus Method , page 166)
Support method to determine whether the tree window can receive the input focus.

CanShowDragImage(see TBaseVirtualTree.CanShowDragImage Method , page 166)
Determines whether a drag image should be shown.

Change(see TBaseVirtualTree.Change Method , page 167)
Central method called when a node's selection state changes.

ChangeScale(see TBaseVirtualTree.ChangeScale Method , page 167)
Helper method called by the VCL when control resizing is due.

CheckParentCheckState(see TBaseVirtualTree.CheckParentCheckState Method , page 167)
Helper method for recursive check state changes.

Clear(see TBaseVirtualTree.Clear Method , page 168)
Clears the tree and removes all nodes.

ClearChecked(see TBaseVirtualTree.ClearChecked Method , page 168)
Not documented.

ClearSelection(see TBaseVirtualTree.ClearSelection Method , page 168)
Removes all nodes from the current selection.

ClearTempCache(see TBaseVirtualTree.ClearTempCache Method , page 168)
Helper method to clear(see TBaseVirtualTree.Clear Method , page 168) the internal temporary node cache.

ColumnIsEmpty(see TBaseVirtualTree.ColumnIsEmpty Method , page 169)
Used to determine if a cell is considered as being empty.

CopyTo(see TBaseVirtualTree.CopyTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 169)
Copies Source and all its child nodes to Target.

CopyToClipBoard(see TBaseVirtualTree.CopyToClipBoard Method , page 169)
Copies all currently selected nodes to the clipboard.

CountLevelDifference(see TBaseVirtualTree.CountLevelDifference Method , page 170)
Determines the level difference of two nodes.

CountVisibleChildren(see TBaseVirtualTree.CountVisibleChildren Method , page 170)
Determines the number of visible child nodes of the given node.

Create(see TBaseVirtualTree.Create Constructor , page 170)
Constructor of the control

CreateParams(see TBaseVirtualTree.CreateParams Method , page 171)
Prepares the creation of the controls window handle.

CreateWnd(see TBaseVirtualTree.CreateWnd Method , page 171)
Initializes data, which depends on the window handle.

CutToClipBoard(see TBaseVirtualTree.CutToClipBoard Method , page 171)
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties(see TBaseVirtualTree.DefineProperties Method , page 171)
Helper method to customize loading and saving persistent tree data.

DeleteChildren(see TBaseVirtualTree.DeleteChildren Method , page 172)
Removes all child nodes from the given node.

DeleteNode(see TBaseVirtualTree.DeleteNode Method , page 172)
Removes the given node from the tree.

DeleteSelectedNodes(see TBaseVirtualTree.DeleteSelectedNodes Method , page 172)
Removes all currently selected nodes form the tree.

Destroy(see TBaseVirtualTree.Destroy Destructor , page 173)
Destructor of the control.

DetermineHiddenChildrenFlag(see TBaseVirtualTree.DetermineHiddenChildrenFlag Method , page 173)
Determines whether all children of a given node are hidden.

DetermineHiddenChildrenFlagAllNodes(see TBaseVirtualTree.DetermineHiddenChildrenFlagAllNodes Method , page 173)
Determines whether all children of all nodes are hidden.

DetermineHitPositionLTR(see TBaseVirtualTree.DetermineHitPositionLTR Method , page 174)
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineHitPositionRTL(see TBaseVirtualTree.DetermineHitPositionLTR Method , page 174)
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineNextCheckState(see TBaseVirtualTree.DetermineNextCheckState Method , page 174)
Not documented.

DetermineScrollDirections(see TBaseVirtualTree.DetermineScrollDirections Method , page 174)
Not documented.

DoAdvancedHeaderDraw(see TBaseVirtualTree.DoAdvancedHeaderDraw Method , page 174)
Not documented.

DoAfterCellPaint(see TBaseVirtualTree.DoAfterCellPaint Method , page 175)
Not documented.

DoAfterItemErase(see TBaseVirtualTree.DoAfterItemErase Method , page 175)
Not documented.

DoAfterItemPaint(see TBaseVirtualTree.DoAfterItemPaint Method , page 175)
Not documented.

DoAfterPaint(see TBaseVirtualTree.DoAfterPaint Method , page 175)
Not documented.

DoAutoScroll(see TBaseVirtualTree.DoAutoScroll Method , page 176)
Enables or disables the auto scroll timer.

DoBeforeCellPaint(see TBaseVirtualTree.DoBeforeCellPaint Method , page 176)
Not documented.

DoBeforeDrag(see TBaseVirtualTree.DoBeforeDrag Method , page 176)
Not documented.

DoBeforeItemErase(see TBaseVirtualTree.DoBeforeItemErase Method , page 177)

Not documented.

DoBeforeItemPaint(see TBaseVirtualTree.DoBeforeItemPaint Method , page 177)
Not documented.

DoBeforePaint(see TBaseVirtualTree.DoBeforePaint Method , page 177)
Not documented.

DoCancelEdit(see TBaseVirtualTree.DoCancelEdit Method , page 177)
Called when the tree should stop editing without accepting changed values.

DoCanEdit(see TBaseVirtualTree.DoCanEdit Method , page 178)
Not documented.

DoChange(see TBaseVirtualTree.DoChange Method , page 178)
Not documented.

DoCheckClick(see TBaseVirtualTree.DoCheckClick Method , page 178)
Not documented.

DoChecked(see TBaseVirtualTree.DoChecked Method , page 178)
Not documented.

DoChecking(see TBaseVirtualTree.DoChecking Method , page 179)
Not documented.

DoCollapsed(see TBaseVirtualTree.DoCollapsed Method , page 179)
Not documented.

DoCollapsing(see TBaseVirtualTree.DoCollapsing Method , page 179)
Not documented.

DoColumnClick(see TBaseVirtualTree.DoColumnClick Method , page 179)
Not documented.

DoColumnDblClick(see TBaseVirtualTree.DoColumnDblClick Method , page 180)
Not documented.

DoColumnResize(see TBaseVirtualTree.DoColumnResize Method , page 180)
Not documented.

DoCompare(see TBaseVirtualTree.DoCompare Method , page 180)
Not documented.

DoCreateDataObject(see TBaseVirtualTree.DoCreateDataObject Method , page 180)
Not documented.

DoCreateDragManager(see TBaseVirtualTree.DoCreateDragManager Method , page 181)
Not documented.

DoCreateEditor(see TBaseVirtualTree.DoCreateEditor Method , page 181)
Not documented.

DoDragDrop(see TBaseVirtualTree.DoDragDrop Method , page 181)
Not documented.

DoDragExpand(see TBaseVirtualTree.DoDragExpand Method , page 181)
Not documented.

DoDragging(see TBaseVirtualTree.DoDragging Method , page 182)
Internal method which handles drag' drop.

DoDragOver(see TBaseVirtualTree.DoDragOver Method , page 182)
Not documented.

DoEdit(see TBaseVirtualTree.DoEdit Method , page 182)
Initiates editing of the currently set focused column and edit node.

DoEndDrag(see TBaseVirtualTree.DoEndDrag Method , page 182)
Not documented.

DoEndEdit(see TBaseVirtualTree.DoEndEdit Method , page 183)
Stops the current edit operation and takes over the new content.

DoExpanded(see TBaseVirtualTree.DoExpanded Method , page 183)
Not documented.

DoExpanding(see TBaseVirtualTree.DoExpanding Method , page 183)
Not documented.

DoFocusChange(see TBaseVirtualTree.DoFocusChange Method , page 184)
Not documented.

DoFocusChanging(see TBaseVirtualTree.DoFocusChanging Method , page 184)
Not documented.

DoFocusNode(see TBaseVirtualTree.DoFocusNode Method , page 184)
Internal method to set the focused node.

DoFreeNode(see TBaseVirtualTree.DoFreeNode Method , page 184)
Not documented.

DoGetAnimationType(see TBaseVirtualTree.DoGetAnimationType Method , page 185)
Determines the type of animation to be used.

DoGetCursor(see TBaseVirtualTree.DoGetCursor Method , page 185)
Not documented.

DoGetHeaderCursor(see TBaseVirtualTree.DoGetHeaderCursor Method , page 185)
Not documented.

DoGetImageIndex(see TBaseVirtualTree.DoGetImageIndex Method , page 186)
Not documented.

DoGetLineStyle(see TBaseVirtualTree.DoGetLineStyle Method , page 186)
Not documented.

DoGetNodeHint(see TBaseVirtualTree.DoGetNodeHint Method , page 186)
Not documented.

DoGetNodeTooltip(see TBaseVirtualTree.DoGetNodeTooltip Method , page 186)
Not documented.

DoGetNodeWidth(see TBaseVirtualTree.DoGetNodeWidth Method , page 187)
Overridable method which always retuns 0.

DoGetPopupMenu(see TBaseVirtualTree.DoGetPopupMenu Method , page 187)
Overridable method which triggers the OnGetPopup event.

DoGetUserClipboardFormats(see TBaseVirtualTree.DoGetUserClipboardFormats Method , page 187)
Not documented.

DoHeaderClick(see TBaseVirtualTree.DoHeaderClick Method , page 187)
Not documented.

DoHeaderDblClick(see TBaseVirtualTree.DoHeaderDblClick Method , page 188)
Not documented.

DoHeaderDragged(see TBaseVirtualTree.DoHeaderDragged Method , page 188)
Not documented.

DoHeaderDraggedOut(see TBaseVirtualTree.DoHeaderDraggedOut Method , page 188)
Not documented.

DoHeaderDragging(see TBaseVirtualTree.DoHeaderDragging Method , page 189)
Not documented.

DoHeaderDraw(see TBaseVirtualTree.DoHeaderDraw Method , page 189)
Not documented.

DoHeaderDrawQueryElements(see TBaseVirtualTree.DoHeaderDrawQueryElements Method , page 189)
Not documented.

DoHeaderMouseDown(see TBaseVirtualTree.DoHeaderMouseDown Method , page 189)
Not documented.

DoHeaderMouseMove(see TBaseVirtualTree.DoHeaderMouseMove Method , page 190)
Not documented.

DoHeaderMouseUp(see TBaseVirtualTree.DoHeaderMouseUp Method , page 190)
Not documented.

DoHotChange(see TBaseVirtualTree.DoHotChange Method , page 190)

Not documented.

DoIncrementalSearch(see TBaseVirtualTree.DoIncrementalSearch Method , page 190)
Not documented.

DoInitChildren(see TBaseVirtualTree.DoInitChildren Method , page 191)
Not documented.

DoInitNode(see TBaseVirtualTree.DoInitNode Method , page 191)
Not documented.

DoKeyAction(see TBaseVirtualTree.DoKeyAction Method , page 191)
Not documented.

DoLoadUserData(see TBaseVirtualTree.DoLoadUserData Method , page 191)
Not documented.

DoMeasureItem(see TBaseVirtualTree.DoMeasureItem Method , page 192)
Not documented.

DoNodeCopied(see TBaseVirtualTree.DoNodeCopied Method , page 192)
Not documented.

DoNodeCopying(see TBaseVirtualTree.DoNodeCopying Method , page 192)
Not documented.

DoNodeMoved(see TBaseVirtualTree.DoNodeMoved Method , page 192)
Not documented.

DoNodeMoving(see TBaseVirtualTree.DoNodeMoving Method , page 193)
Not documented.

DoPaintBackground(see TBaseVirtualTree.DoPaintBackground Method , page 193)
Not documented.

DoPaintDropMark(see TBaseVirtualTree.DoPaintDropMark Method , page 193)
Overridable method which draws the small line on top of a nodes image depending on the current drop state.

DoPaintNode(see TBaseVirtualTree.DoPaintNode Method , page 193)
Overridable method which does nothing.

DoPopupMenu(see TBaseVirtualTree.DoPopupMenu Method , page 194)
Overridable method which shows the popup menu for the given node.

DoRenderOLEData(see TBaseVirtualTree.DoRenderOLEData Method , page 194)
Not documented.

DoReset(see TBaseVirtualTree.DoReset Method , page 194)
Not documented.

DoSaveUserData(see TBaseVirtualTree.DoSaveUserData Method , page 194)
Not documented.

DoScroll(see TBaseVirtualTree.DoScroll Method , page 195)
Overridable method which triggers the OnScroll(see TBaseVirtualTree.OnScroll Event, page 152) event.

DoSetOffsetXY(see TBaseVirtualTree.DoSetOffsetXY Method , page 195)
Internal core routine to set the tree's scroll position.

DoShowScrollbar(see TBaseVirtualTree.DoShowScrollbar Method , page 195)
Not documented.

DoStartDrag(see TBaseVirtualTree.DoStartDrag Method , page 196)
Not documented.

DoStateChange(see TBaseVirtualTree.DoStateChange Method , page 196)
Not documented.

DoStructureChange(see TBaseVirtualTree.DoStructureChange Method , page 196)
Not documented.

DoTimerScroll(see TBaseVirtualTree.DoTimerScroll Method , page 196)
Callback method which is triggered whenever the scroll timer fires.

DoUpdating(see TBaseVirtualTree.DoUpdating Method , page 197)
Not documented.

DoValidateCache(see TBaseVirtualTree.DoValidateCache Method , page 197)
Not documented.

DragCanceled(see TBaseVirtualTree.DragCanceled Method , page 197)
Called by the VCL when a drag'n drop operation was canceled by the user.

DragDrop(see TBaseVirtualTree.DragDrop Method , page 197)
Helper method, which is used when a drag operation is finished.

DragEnter(see TBaseVirtualTree.DragEnter Method , page 198)
Not documented.

DragFinished(see TBaseVirtualTree.DragFinished Method , page 198)
Called when a drag operation is finished (accepted or cancelled).

Dragging(see TBaseVirtualTree.Dragging Method , page 198)
Returns true if a drag'n drop operation is in progress.

DragLeave(see TBaseVirtualTree.DragLeave Method , page 199)
Not documented.

DragOver(see TBaseVirtualTree.DragOver Method , page 199)
Not documented.

DrawDottedHLine(see TBaseVirtualTree.DrawDottedHLine Method , page 199)
Not documented.

DrawDottedVLine(see TBaseVirtualTree.DrawDottedVLine Method , page 199)
Not documented.

EditNode(see TBaseVirtualTree.EditNode Method , page 200)
Starts editing the given node if allowed to.

EndEditNode(see TBaseVirtualTree.EndEditNode Method , page 200)
Stops node editing if it was started before.

EndSynch(see TBaseVirtualTree.EndSynch Method , page 200)
Counterpart to BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164).

EndUpdate(see TBaseVirtualTree.EndUpdate Method , page 201)
Resets the update lock set by BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164).

ExecuteAction(see TBaseVirtualTree.ExecuteAction Method , page 201)
Not documented.

FindNodeInSelection(see TBaseVirtualTree.FindNodeInSelection Method , page 201)
Helper method to find the given node in the current selection.

FinishChunkHeader(see TBaseVirtualTree.FinishChunkHeader Method , page 201)
Not documented.

FinishCutOrCopy(see TBaseVirtualTree.FinishCutOrCopy Method , page 202)
Stops any pending cut or copy clipboard operation.

FlushClipboard(see TBaseVirtualTree.FlushClipboard Method , page 202)
Renders all pending clipboard data.

FontChanged(see TBaseVirtualTree.FontChanged Method , page 202)
Not documented.

FullCollapse(see TBaseVirtualTree.FullCollapse Method , page 203)
Collapses all nodes in the tree.

FullExpand(see TBaseVirtualTree.FullExpand Method , page 203)
Expands all nodes in the tree.

GetBorderDimensions(see TBaseVirtualTree.GetBorderDimensions Method , page 203)
Not documented.

GetCheckImage(see TBaseVirtualTree.GetCheckImage Method , page 203)
Not documented.

GetCheckImageListFor(see TBaseVirtualTree.GetCheckImageListFor Method , page 204)
Not documented.

GetColumnClass(see TBaseVirtualTree.GetColumnClass Method , page 204)

Returns the class to be used to manage columns in the tree.

GetControlsAlignment(see TBaseVirtualTree.GetControlsAlignment Method , page 204)
Not documented.

GetDisplayRect(see TBaseVirtualTree.GetDisplayRect Method , page 205)
Returns the visible region used by the given node in client coordinates.

GetFirst(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstChecked(see TBaseVirtualTree.GetFirstChecked Method , page 206)
Not documented.

GetFirstChild(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstCutCopy(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstInitialized(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstSelected(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisible(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleChild(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleChildNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetHeaderClass(see TBaseVirtualTree.GetHeaderClass Method , page 206)
Returns the header class to be used by the tree.

GetHintWindowClass(see TBaseVirtualTree.GetHintWindowClass Method , page 206)
Not documented.

GetHitTestInfoAt(see TBaseVirtualTree.GetHitTestInfoAt Method , page 206)
Returns information about the node at the given position.

GetImageIndex(see TBaseVirtualTree.GetImageIndex Method , page 207)
Not documented.

GetLast(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastChild(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastChildNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastInitialized(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisible(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleChild(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleChildNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetMaxColumnWidth(see TBaseVirtualTree.GetMaxColumnWidth Method , page 208)
Returns the width of the largest node in the given column.

GetMaxRightExtend(see TBaseVirtualTree.GetMaxRightExtend Method , page 208)
Determines the maximum with of the currently visible part of the tree.

GetNativeClipboardFormats(see TBaseVirtualTree.GetNativeClipboardFormats Method , page 208)
Used to let descendants and the application add their own supported clipboard formats.

GetNext(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextChecked(see TBaseVirtualTree.GetNextChecked Method , page 209)
Not documented.

GetNextCutCopy(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextInitialized(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextSelected(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextSibling(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisible(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleSibling(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleSiblingNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNodeAt(see TBaseVirtualTree.GetNodeAt Method (Integer, Integer), page 210)
Not documented.

GetNodeData(see TBaseVirtualTree.GetNodeData Method , page 210)
Returns the address of the user data area of the given node.

GetNodeLevel(see TBaseVirtualTree.GetNodeLevel Method , page 210)
Returns the indentation level of the given node.

GetOptionsClass(see TBaseVirtualTree.GetOptionsClass Method , page 211)
Customization helper to determine which options class the tree should use.

GetPrevious(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousInitialized(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousSibling(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisible(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleSibling(see TBaseVirtualTree.GetPrevious Method , page 212)

Group of node navigation functions.

GetPreviousVisibleSiblingNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetSortedCutCopySet(see TBaseVirtualTree.GetSortedCutCopySet Method , page 212)
Returns a sorted list of nodes, which are marked for s cut or copy clipboard operation.

GetSortedSelection(see TBaseVirtualTree.GetSortedSelection Method , page 213)
Returns a sorted list of all currently selected nodes.

GetTextInfo(see TBaseVirtualTree.GetTextInfo Method , page 213)
Helper method for node editors, hints etc.

GetTreeFromDataObject(see TBaseVirtualTree.GetTreeFromDataObject Method , page 213)
OLE drag'n drop and clipboard support method.

GetTreeRect(see TBaseVirtualTree.GetTreeRect Method , page 214)
Returns the size of the virtual tree image.

GetVisibleParent(see TBaseVirtualTree.GetVisibleParent Method , page 214)
Returns the first (nearest) parent node, which is visible.

HandleHotTrack(see TBaseVirtualTree.HandleHotTrack Method , page 214)
Not documented.

HandleIncrementalSearch(see TBaseVirtualTree.HandleIncrementalSearch Method , page 214)
Not documented.

HandleMouseDblClick(see TBaseVirtualTree.HandleMouseDblClick Method , page 215)
Not documented.

HandleMouseDown(see TBaseVirtualTree.HandleMouseDown Method , page 215)
Not documented.

HandleMouseUp(see TBaseVirtualTree.HandleMouseUp Method , page 215)
Not documented.

HasAsParent(see TBaseVirtualTree.HasAsParent Method , page 215)
Determines if the given node has got another node as one of its parents.

HasImage(see TBaseVirtualTree.HasImage Method , page 216)
Not documented.

HasPopupMenu(see TBaseVirtualTree.HasPopupMenu Method , page 216)
Determines whether there is a pop up menu assigned to the tree.

InitChildren(see TBaseVirtualTree.InitChildren Method , page 216)
Not documented.

InitNode(see TBaseVirtualTree.InitNode Method , page 217)
Not documented.

InsertNode(see TBaseVirtualTree.InsertNode Method , page 217)
Inserts a new node and returns it to the caller.

InternalAddFromStream(see TBaseVirtualTree.InternalAddFromStream Method , page 217)
Not documented.

InternalAddToSelection(see TBaseVirtualTree.InternalAddToSelection Method (PVirtualNode, Boolean), page 218)
Not documented.

InternalCacheNode(see TBaseVirtualTree.InternalCacheNode Method , page 218)
Not documented.

InternalClearSelection(see TBaseVirtualTree.InternalClearSelection Method , page 218)
Not documented.

InternalConnectNode(see TBaseVirtualTree.InternalConnectNode Method , page 219)
Not documented.

InternalData(see TBaseVirtualTree.InternalData Method , page 219)
Returns the address of the internal data for a tree class.

InternalDisconnectNode(see TBaseVirtualTree.InternalDisconnectNode Method , page 219)
Not documented.

InternalRemoveFromSelection(see TBaseVirtualTree.InternalRemoveFromSelection Method , page 220)
Not documented.

InvalidateCache(see TBaseVirtualTree.InvalidateCache Method , page 220)
Empties the internal node cache and marks it as invalid.

InvalidateChildren(see TBaseVirtualTree.InvalidateChildren Method , page 220)
Invalidates all children of the given node.

InvalidateColumn(see TBaseVirtualTree.InvalidateColumn Method , page 220)
Invalidates the client area part of a column.

InvalidateNode(see TBaseVirtualTree.InvalidateNode Method , page 221)
Invalidates the given node.

InvalidateToBottom(see TBaseVirtualTree.InvalidateToBottom Method , page 221)
Invalidates the client area starting with the top position of the given node.

InvertSelection(see TBaseVirtualTree.InvertSelection Method , page 221)
Inverts the current selection.

IsEditing(see TBaseVirtualTree.IsEditing Method , page 222)
Tells the caller whether the tree is currently in edit mode.

IsMouseSelecting(see TBaseVirtualTree.IsMouseSelecting Method , page 222)
Tell the caller whether the tree is currently in draw selection mode.

IterateSubtree(see TBaseVirtualTree.IterateSubtree Method , page 222)
Iterator method to go through all nodes of a given sub tree.

Loaded(see TBaseVirtualTree.Loaded Method , page 223)
Not documented.

LoadFromFile(see TBaseVirtualTree.LoadFromFile Method , page 223)
Loads previously streamed out tree data back in again.

LoadFromStream(see TBaseVirtualTree.LoadFromFile Method , page 223)
Loads previously streamed out tree data back in again.

MainColumnChanged(see TBaseVirtualTree.MainColumnChanged Method , page 223)
Not documented.

MarkCutCopyNodes(see TBaseVirtualTree.MarkCutCopyNodes Method , page 223)
Not documented.

MeasureItemHeight(see TBaseVirtualTree.MeasureItemHeight Method , page 224)
Not documented.

MouseMove(see TBaseVirtualTree.MouseMove Method , page 224)
Not documented.

MoveTo(see TBaseVirtualTree.MoveTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 224)
Moves Source and all its child nodes to Target.

Notification(see TBaseVirtualTree.Notification Method , page 225)
Not documented.

OriginalWMNCPaint(see TBaseVirtualTree.OriginalWMNCPaint Method , page 225)
Not documented.

Paint(see TBaseVirtualTree.Paint Method , page 225)
TControl's Paint method used here to display the tree.

PaintCheckImage(see TBaseVirtualTree.PaintCheckImage Method , page 225)
Not documented.

PaintImage(see TBaseVirtualTree.PaintImage Method , page 226)
Not documented.

PaintNodeButton(see TBaseVirtualTree.PaintNodeButton Method , page 226)
Not documented.

PaintSelectionRectangle(see TBaseVirtualTree.PaintSelectionRectangle Method , page 226)
Not documented.

PaintTree(see TBaseVirtualTree.PaintTree Method , page 226)

Main paint routine for the tree image.

PaintTreeLines(see TBaseVirtualTree.PaintTreeLines Method , page 227)
Not documented.

PanningWindowProc(see TBaseVirtualTree.PanningWindowProc Method , page 227)
Not documented.

PasteFromClipboard(see TBaseVirtualTree.PasteFromClipboard Method , page 227)
Inserts the content of the clipboard into the tree.

PrepareDragImage(see TBaseVirtualTree.PrepareDragImage Method , page 228)
Not documented.

Print(see TBaseVirtualTree.Print Method , page 228)
Not documented.

ProcessDrop(see TBaseVirtualTree.ProcessDrop Method , page 228)
Helper method to ease OLE drag'n drop operations.

ProcessOLEData(see TBaseVirtualTree.ProcessOLEData Method , page 229)
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk(see TBaseVirtualTree.ReadChunk Method , page 229)
Not documented.

ReadNode(see TBaseVirtualTree.ReadNode Method , page 229)
Not documented.

RedirectFontChangeEvent(see TBaseVirtualTree.RedirectFontChangeEvent Method , page 230)
Not documented.

ReinitChildren(see TBaseVirtualTree.ReinitChildren Method , page 230)
Forces all child nodes of Node to be reinitialized.

ReinitNode(see TBaseVirtualTree.ReinitNode Method , page 230)
Forces a reinitialization of the given node.

RemoveFromSelection(see TBaseVirtualTree.RemoveFromSelection Method , page 230)
Removes the given node from the current selection.

RenderOLEData(see TBaseVirtualTree.RenderOLEData Method , page 231)
Renders pending OLE data.

RepaintNode(see TBaseVirtualTree.RepaintNode Method , page 231)
Causes the treeview to repaint the given node.

ResetNode(see TBaseVirtualTree.ResetNode Method , page 231)
Resets the given node to uninitialized.

ResetRangeAnchor(see TBaseVirtualTree.ResetRangeAnchor Method , page 232)
Not documented.

RestoreFontChangeEvent(see TBaseVirtualTree.RestoreFontChangeEvent Method , page 232)
Not documented.

SaveToFile(see TBaseVirtualTree.SaveToFile Method , page 232)
Saves the entire content of the tree into a file or stream.

SaveToStream(see TBaseVirtualTree.SaveToFile Method , page 232)
Saves the entire content of the tree into a file or stream.

ScrollIntoView(see TBaseVirtualTree.ScrollIntoView Method , page 232)
Scrolls the tree so that the given node comes in the client area.

SelectAll(see TBaseVirtualTree.SelectAll Method , page 233)
Selects all nodes in the tree.

SelectNodes(see TBaseVirtualTree.SelectNodes Method , page 233)
Selects a range of nodes.

SetBiDiMode(see TBaseVirtualTree.SetBiDiMode Method , page 233)
Not documented.

SetFocusedNodeAndColumn(see TBaseVirtualTree.SetFocusedNodeAndColumn Method , page 234)
Not documented.

SkipNode(see TBaseVirtualTree.SkipNode Method , page 234)
Not documented.

Sort(see TBaseVirtualTree.Sort Method , page 234)
Sorts the given node.

SortTree(see TBaseVirtualTree.SortTree Method , page 234)
Sorts the entire tree view.

StartWheelPanning(see TBaseVirtualTree.StartWheelPanning Method , page 235)
Not documented.

StopWheelPanning(see TBaseVirtualTree.StopWheelPanning Method , page 235)
Not documented.

StructureChange(see TBaseVirtualTree.StructureChange Method , page 235)
Not documented.

SuggestDropEffect(see TBaseVirtualTree.SuggestDropEffect Method , page 236)
Not documented.

ToggleNode(see TBaseVirtualTree.ToggleNode Method , page 236)
Changes a node's expand state to the opposite state.

ToggleSelection(see TBaseVirtualTree.ToggleSelection Method , page 236)
Toggles the selection state of a range of nodes.

UnselectNodes(see TBaseVirtualTree.UnselectNodes Method , page 236)
Deselects a range of nodes.

UpdateAction(see TBaseVirtualTree.UpdateAction Method , page 237)
Not documented.

UpdateDesigner(see TBaseVirtualTree.UpdateDesigner Method , page 237)
Not documented.

UpdateEditBounds(see TBaseVirtualTree.UpdateEditBounds Method , page 237)
Not documented.

UpdateHeaderRect(see TBaseVirtualTree.UpdateHeaderRect Method , page 237)
Not documented.

UpdateHorizontalScrollBar(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateScrollBars(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateVerticalScrollBar(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateWindowAndDragImage(see TBaseVirtualTree.UpdateWindowAndDragImage Method , page 238)
Not documented.

UseRightToLeftReading(see TBaseVirtualTree.UseRightToLeftReading Method , page 238)
Helper method for right-to-left layout.

ValidateCache(see TBaseVirtualTree.ValidateCache Method , page 239)
Initiates the validation of the internal node cache.

ValidateChildren(see TBaseVirtualTree.ValidateChildren Method , page 239)
Validates all children of a given node.

ValidateNode(see TBaseVirtualTree.ValidateNode Method , page 239)
Validates a given node.

ValidateNodeDataSize(see TBaseVirtualTree.ValidateNodeDataSize Method , page 239)
Helper method for node data size initalization.

WndProc(see TBaseVirtualTree.WndProc Method , page 240)
Redirected window procedure to do some special processing.

WriteChunks(see TBaseVirtualTree.WriteChunks Method , page 240)
Writes the core chunks for the given node to the given stream.

WriteNode(see TBaseVirtualTree.WriteNode Method , page 241)

Writes the cover (envelop) chunk for the given node to the given stream.

Properties

Action(see TVirtualDrawTree.Action Property, page 349)
Not documented.

Align(see TVirtualDrawTree.Align Property, page 349)
Not documented.

Alignment(see TVirtualDrawTree.Alignment Property, page 350)
Determines the horizontal alignment of text if no columns are defined.

Anchors(see TVirtualDrawTree.Anchors Property, page 350)
Not documented.

AnimationDuration(see TVirtualDrawTree.AnimationDuration Property, page 350)
Determines the maximum duration the tree can use to play an animation.

AutoExpandDelay(see TVirtualDrawTree.AutoExpandDelay Property, page 351)
Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay(see TVirtualDrawTree.AutoScrollDelay Property, page 351)
Time which determines when auto scrolling should start.

AutoScrollInterval(see TVirtualDrawTree.AutoScrollInterval Property, page 351)
Time interval between scroll events when doing auto scroll.

Background(see TVirtualDrawTree.Background Property, page 351)
Holds a background image for the tree.

BackgroundOffsetX(see TVirtualDrawTree.BackgroundOffsetX Property, page 352)
Horizontal offset of the background image.

BackgroundOffsetY(see TVirtualDrawTree.BackgroundOffsetY Property, page 352)
Vertical offset of the background image.

BevelEdges(see TVirtualDrawTree.BevelEdges Property, page 352)
Not documented.

BevelInner(see TVirtualDrawTree.BevelInner Property, page 352)
Not documented.

BevelKind(see TVirtualDrawTree.BevelKind Property, page 353)
Not documented.

BevelOuter(see TVirtualDrawTree.BevelOuter Property, page 353)
Not documented.

BevelWidth(see TVirtualDrawTree.BevelWidth Property, page 353)
Not documented.

BiDiMode(see TVirtualDrawTree.BiDiMode Property, page 353)
Not documented.

BorderStyle(see TVirtualDrawTree.BorderStyle Property, page 354)
Same as TForm.BorderStyle.

BorderWidth(see TVirtualDrawTree.BorderWidth Property, page 354)
Not documented.

ButtonFillMode(see TVirtualDrawTree.ButtonFillMode Property, page 354)
Determines how to fill the background of the node buttons.

ButtonStyle(see TVirtualDrawTree.ButtonStyle Property, page 355)
Determines the look of node buttons.

Canvas(see TVirtualDrawTree.Canvas Property, page 355)
Not documented.

ChangeDelay(see TVirtualDrawTree.ChangeDelay Property, page 355)
Time which determines when the OnChange event should be triggered after the actual change event.

CheckImageKind(see TVirtualDrawTree.CheckImageKind Property, page 355)
Determines which images should be used for checkboxes and radio buttons.

ClipboardFormats(see TVirtualDrawTree.ClipboardFormats Property, page 356)

Special class to keep a list of clipboard format descriptions.

Color(see TVirtualDrawTree.Color Property, page 356)
Not documented.

Colors(see TVirtualDrawTree.Colors Property, page 356)
A collection of colors used in the tree.

Constraints(see TVirtualDrawTree.Constraints Property, page 357)
Not documented.

Ctl3D(see TVirtualDrawTree.Ctl3D Property, page 357)
Not documented.

CustomCheckImages(see TVirtualDrawTree.CustomCheckImages Property, page 357)
Assign your own image list to get the check images you like most.

DefaultNodeHeight(see TVirtualDrawTree.DefaultNodeHeight Property, page 357)
Read or set the height new nodes get as initial value.

DefaultPasteMode(see TVirtualDrawTree.DefaultPasteMode Property, page 358)
Read or set the value, which determines where to add pasted nodes to.

DragCursor(see TVirtualDrawTree.DragCursor Property, page 358)
Not documented.

DragHeight(see TVirtualDrawTree.DragHeight Property, page 358)
Read or set the vertical limit of the internal drag image.

DragImageKind(see TVirtualDrawTree.DragImageKind Property, page 359)
Read or set what should be shown in the drag image.

DragKind(see TVirtualDrawTree.DragKind Property, page 359)
Not documented.

DragMode(see TVirtualDrawTree.DragMode Property, page 359)
Not documented.

DragOperations(see TVirtualDrawTree.DragOperations Property, page 359)
Read or set which drag operations may be allowed in the tree.

DragType(see TVirtualDrawTree.DragType Property, page 360)
Read or set which subsystem should be used for dragging.

DragWidth(see TVirtualDrawTree.DragWidth Property, page 360)
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode(see TVirtualDrawTree.DrawSelectionMode Property, page 360)
Read or set how multiselection with the mouse is to be visualized.

EditDelay(see TVirtualDrawTree.EditDelay Property, page 361)
Read or set the maximum time between two single clicks on the same node, which should start node editing.

Enabled(see TVirtualDrawTree.Enabled Property, page 361)
Not documented.

Font(see TVirtualDrawTree.Font Property, page 361)
Same as TWinControl.Font.

Header(see TVirtualDrawTree.Header Property, page 361)
Provides access to the header instance.

HintAnimation(see TVirtualDrawTree.HintAnimation Property, page 362)
Read or set the current hint animation type.

HintMode(see TVirtualDrawTree.HintMode Property, page 362)
Read or set what type of hint you want for the tree view.

HotCursor(see TVirtualDrawTree.HotCursor Property, page 362)
Read or set which cursor should be used for hot nodes.

Images(see TVirtualDrawTree.Images Property, page 363)
Read or set the tree's normal image list.

IncrementalSearch(see TVirtualDrawTree.IncrementalSearch Property, page 363)
Read or set the current incremental search mode.

IncrementalSearchDirection(see TVirtualDrawTree.IncrementalSearchDirection Property, page 363)
Read or set the direction to be used for incremental search.

IncrementalSearchStart(see TVirtualDrawTree.IncrementalSearchStart Property, page 364)
Read or set where to start incremental search.

IncrementalSearchTimeout(see TVirtualDrawTree.IncrementalSearchTimeout Property, page 364)
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Indent(see TVirtualDrawTree.Indent Property, page 364)
Read or set the indentation amount for node levels.

LineMode(see TVirtualDrawTree.LineMode Property, page 365)
Read or set the mode of the tree lines.

LineStyle(see TVirtualDrawTree.LineStyle Property, page 365)
Read or set the mode of the tree lines.

Margin(see TVirtualDrawTree.Margin Property, page 365)
Read or set the tree's node margin.

NodeAlignment(see TVirtualDrawTree.NodeAlignment Property, page 366)
Read or set the node alignment value.

NodeDataSize(see TVirtualDrawTree.NodeDataSize Property, page 366)
Read or set the extra data size for each node.

OnClick(see TVirtualDrawTree.OnClick Property, page 370)
Not documented.

OnDblClick(see TVirtualDrawTree.OnDblClick Property, page 373)
Not documented.

OnEndDock(see TVirtualDrawTree.OnEndDock Property, page 377)
Not documented.

OnEndDrag(see TVirtualDrawTree.OnEndDrag Property, page 377)
Not documented.

OnEnter(see TVirtualDrawTree.OnEnter Property, page 377)
Not documented.

OnExit(see TVirtualDrawTree.OnExit Property, page 378)
Not documented.

OnKeyDown(see TVirtualDrawTree.OnKeyDown Property, page 388)
Not documented.

OnKeyPress(see TVirtualDrawTree.OnKeyPress Property, page 389)
Not documented.

OnKeyUp(see TVirtualDrawTree.OnKeyUp Property, page 389)
Not documented.

OnMouseDown(see TVirtualDrawTree.OnMouseDown Property, page 390)
Not documented.

OnMouseMove(see TVirtualDrawTree.OnMouseMove Property, page 390)
Not documented.

OnMouseUp(see TVirtualDrawTree.OnMouseUp Property, page 391)
Not documented.

OnMouseWheel(see TVirtualDrawTree.OnMouseWheel Property, page 391)
Not documented.

OnResize(see TVirtualDrawTree.OnResize Property, page 393)
Not documented.

OnStartDock(see TVirtualDrawTree.OnStartDock Property, page 394)
Not documented.

ParentBiDiMode(see TVirtualDrawTree.ParentBiDiMode Property, page 395)
Not documented.

ParentColor(see TVirtualDrawTree.ParentColor Property, page 396)

Not documented.

ParentCtl3D(see TVirtualDrawTree.ParentCtl3D Property, page 396)
Not documented.

ParentFont(see TVirtualDrawTree.ParentFont Property, page 396)
Not documented.

ParentShowHint(see TVirtualDrawTree.ParentShowHint Property, page 396)
Not documented.

PopupMenu(see TVirtualDrawTree.PopupMenu Property, page 397)
Not documented.

RootNodeCount(see TVirtualDrawTree.RootNodeCount Property, page 397)
Read or set the number of nodes on the top level.

ScrollBarOptions(see TVirtualDrawTree.ScrollBarOptions Property, page 397)
Reference to the scroll bar options class.

SelectionBlendFactor(see TVirtualDrawTree.SelectionBlendFactor Property, page 398)
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

SelectionCurveRadius(see TVirtualDrawTree.SelectionCurveRadius Property, page 398)
Read or set the current corner radius for node selection rectangles.

ShowHint(see TVirtualDrawTree.ShowHint Property, page 398)
Not documented.

StateImages(see TVirtualDrawTree.StateImages Property, page 399)
Reference to the images list which is used for the state images.

TabOrder(see TVirtualDrawTree.TabOrder Property, page 399)
Not documented.

TabStop(see TVirtualDrawTree.TabStop Property, page 399)
Not documented.

TextMargin(see TVirtualDrawTree.TextMargin Property, page 400)
Read or set the distance of the node caption to its borders.

TreeOptions(see TVirtualDrawTree.TreeOptions Property, page 400)
Reference to the tree's options.

Visible(see TVirtualDrawTree.Visible Property, page 400)
Not documented.

WantTabs(see TVirtualDrawTree.WantTabs Property, page 400)
Read or set whether the tree wants to process tabs on its own.

TBaseVirtualTree Class

Alignment(see TBaseVirtualTree.Alignment Property, page 107)
Determines the horizontal alignment of text if no columns are defined.

AnimationDuration(see TBaseVirtualTree.AnimationDuration Property, page 107)
Determines the maximum duration the tree can use to play an animation.

AutoExpandDelay(see TBaseVirtualTree.AutoExpandDelay Property, page 107)
Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay(see TBaseVirtualTree.AutoScrollDelay Property, page 108)
Time which determines when auto scrolling should start.

AutoScrollInterval(see TBaseVirtualTree.AutoScrollInterval Property, page 108)
Time interval between scroll events when doing auto scroll.

Background(see TBaseVirtualTree.Background Property, page 108)
Holds a background image for the tree.

BackgroundOffsetX(see TBaseVirtualTree.BackgroundOffsetX Property, page 109)
Horizontal offset of the background image.

BackgroundOffsetY(see TBaseVirtualTree.BackgroundOffsetY Property, page 109)
Vertical offset of the background image.

BorderStyle(see TBaseVirtualTree.BorderStyle Property, page 109)

Same as TForm.BorderStyle.

ButtonFillMode(see TBaseVirtualTree.ButtonFillMode Property, page 109)
Determines how to fill the background of the node buttons.

ButtonStyle(see TBaseVirtualTree.ButtonStyle Property, page 110)
Determines the look of node buttons.

ChangeDelay(see TBaseVirtualTree.ChangeDelay Property, page 110)
Time which determines when the OnChange(see TBaseVirtualTree.OnChange Event, page 131) event should be triggered after the actual change event.

CheckImageKind(see TBaseVirtualTree.CheckImageKind Property, page 110)
Determines which images should be used for checkboxes and radio buttons.

CheckImages(see TBaseVirtualTree.CheckImages Property, page 111)
Not documented.

CheckState(see TBaseVirtualTree.CheckState Property, page 111)
Read or set the check state of a node.

CheckType(see TBaseVirtualTree.CheckType Property, page 111)
Read or set the check type of a node.

ChildCount(see TBaseVirtualTree.ChildCount Property, page 111)
Read or set the number of child nodes of a node.

ChildrenInitialized(see TBaseVirtualTree.ChildrenInitialized Property, page 112)
Read whether a node's child count has been initialized already.

ClipboardFormats(see TBaseVirtualTree.ClipboardFormats Property, page 112)
Special class to keep a list of clipboard format descriptions.

Colors(see TBaseVirtualTree.Colors Property, page 112)
A collection of colors used in the tree.

CustomCheckImages(see TBaseVirtualTree.CustomCheckImages Property, page 113)
Assign your own image list to get the check images you like most.

DefaultNodeHeight(see TBaseVirtualTree.DefaultNodeHeight Property, page 113)
Read or set the height new nodes get as initial value.

DefaultPasteMode(see TBaseVirtualTree.DefaultPasteMode Property, page 113)
Read or set the value, which determines where to add pasted nodes to.

DragHeight(see TBaseVirtualTree.DragHeight Property, page 114)
Read or set the vertical limit of the internal drag image.

DragImage(see TBaseVirtualTree.DragImage Property, page 114)
Holds the instance of the internal drag image.

DragImageKind(see TBaseVirtualTree.DragImageKind Property, page 114)
Read or set what should be shown in the drag image.

DragManager(see TBaseVirtualTree.DragManager Property, page 114)
Holds the reference to the internal drag manager.

DragOperations(see TBaseVirtualTree.DragOperations Property, page 115)
Read or set which drag operations may be allowed in the tree.

DragSelection(see TBaseVirtualTree.DragSelection Property, page 115)
Keeps a temporary list of nodes during drag'n drop.

DragType(see TBaseVirtualTree.DragType Property, page 115)
Read or set which subsystem should be used for dragging(see TBaseVirtualTree.Dragging Method , page 198).

DragWidth(see TBaseVirtualTree.DragWidth Property, page 116)
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode(see TBaseVirtualTree.DrawSelectionMode Property, page 116)
Read or set how multiselection with the mouse is to be visualized.

DropTargetNode(see TBaseVirtualTree.DropTargetNode Property, page 116)
Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

EditColumn(see TBaseVirtualTree.EditColumn Property, page 117)
Not documented.

EditDelay(see TBaseVirtualTree.EditDelay Property, page 117)
Read or set the maximum time between two single clicks on the same node, which should start node editing.

EditLink(see TBaseVirtualTree.EditLink Property, page 117)
Keeps a reference to the internal edit link during a node edit operation.

Expanded(see TBaseVirtualTree.Expanded Property, page 118)
Read or set the expanded state of a particular node.

FocusedColumn(see TBaseVirtualTree.FocusedColumn Property, page 118)
Read or set the currently focused collumn.

FocusedNode(see TBaseVirtualTree.FocusedNode Property, page 118)
Read or set the currently focused node.

Font(see TBaseVirtualTree.Font Property, page 119)
Same as TWinControl.Font.

FullyVisible(see TBaseVirtualTree.FullyVisible Property, page 119)
Read or set whether a node is fully visible or not.

HasChildren(see TBaseVirtualTree.HasChildren Property, page 119)
Read or set whether a node has got children.

Header(see TBaseVirtualTree.Header Property, page 120)
Provides access to the header instance.

HeaderRect(see TBaseVirtualTree.HeaderRect Property, page 120)
Returns the non-client-area rectangle used for the header.

HintAnimation(see TBaseVirtualTree.HintAnimation Property, page 120)
Read or set the current hint animation type.

HintMode(see TBaseVirtualTree.HintMode Property, page 121)
Read or set what type of hint you want for the tree view.

HotCursor(see TBaseVirtualTree.HotCursor Property, page 121)
Read or set which cursor should be used for hot nodes.

HotNode(see TBaseVirtualTree.HotNode Property, page 121)
Read, which node is currently the hot node.

Images(see TBaseVirtualTree.Images Property, page 122)
Read or set the tree's normal image list.

IncrementalSearch(see TBaseVirtualTree.IncrementalSearch Property, page 122)
Read or set the current incremental search mode.

IncrementalSearchDirection(see TBaseVirtualTree.IncrementalSearchDirection Property, page 122)
Read or set the direction to be used for incremental search.

IncrementalSearchStart(see TBaseVirtualTree.IncrementalSearchStart Property, page 123)
Read or set where to start incremental search.

IncrementalSearchTimeout(see TBaseVirtualTree.IncrementalSearchTimeout Property, page 123)
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Indent(see TBaseVirtualTree.Indent Property, page 123)
Read or set the indentation amount for node levels.

IsDisabled(see TBaseVirtualTree.IsDisabled Property, page 124)
Read or set the enabled state of the given node.

IsVisible(see TBaseVirtualTree.IsVisible Property, page 124)
Read or set the visibility state of the given node.

LastClickPos(see TBaseVirtualTree.LastClickPos Property, page 124)
Used for retained drag start and wheel mouse scrolling.

LastDropMode(see TBaseVirtualTree.LastDropMode Property, page 125)
Read how the last drop operation finished.

LineMode(see TBaseVirtualTree.LineMode Property, page 125)
Read or set the mode of the tree lines.

LineStyle(see TBaseVirtualTree.LineStyle Property, page 125)

Read or set the mode of the tree lines.

Margin(see TBaseVirtualTree.Margin Property, page 125)
Read or set the tree's node margin.

MultiLine(see TBaseVirtualTree.MultiLine Property, page 126)
Read or toggle the multiline feature for a given node.

NodeAlignment(see TBaseVirtualTree.NodeAlignment Property, page 126)
Read or set the node alignment value.

NodeDataSize(see TBaseVirtualTree.NodeDataSize Property, page 127)
Read or set the extra data size for each node.

NodeHeight(see TBaseVirtualTree.NodeHeight Property, page 127)
Read or set a node's height.

NodeParent(see TBaseVirtualTree.NodeParent Property, page 127)
Read or set a node's parent node.

OffsetX(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

OffsetXY(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

OffsetY(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

RootNode(see TBaseVirtualTree.RootNode Property, page 153)
Reference to the internal root node which is the anchor of the entire tree node hierarchy.

RootNodeCount(see TBaseVirtualTree.RootNodeCount Property, page 153)
Read or set the number of nodes on the top level.

ScrollBarOptions(see TBaseVirtualTree.ScrollBarOptions Property, page 154)
Reference to the scroll bar options class.

SearchBuffer(see TBaseVirtualTree.SearchBuffer Property, page 154)
Current input string for incremental search.

Selected(see TBaseVirtualTree.Selected Property, page 154)
Property to modify or determine the selection state of a node.

SelectedCount(see TBaseVirtualTree.SelectedCount Property, page 155)
Contains the number of selected nodes.

SelectionBlendFactor(see TBaseVirtualTree.SelectionBlendFactor Property, page 155)
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

SelectionCurveRadius(see TBaseVirtualTree.SelectionCurveRadius Property, page 155)
Read or set the current corner radius for node selection rectangles.

StateImages(see TBaseVirtualTree.StateImages Property, page 156)
Reference to the images list which is used for the state images.

TextMargin(see TBaseVirtualTree.TextMargin Property, page 156)
Read or set the distance of the node caption to its borders.

TopNode(see TBaseVirtualTree.TopNode Property, page 157)
The top node is the node which is currently at the top border of the client area.

TotalCount(see TBaseVirtualTree.TotalCount Property, page 157)
Returns the number of nodes in the tree.

TotalInternalDataSize(see TBaseVirtualTree.TotalInternalDataSize Property, page 157)
Keeps the currently accumulated data size for one node.

TreeOptions(see TBaseVirtualTree.TreeOptions Property, page 158)
Reference to the tree's options.

TreeStates(see TBaseVirtualTree.TreeStates Property, page 158)
Property which keeps a set of flags which indicate current operation and states of the tree.

UpdateCount(see TBaseVirtualTree.UpdateCount Property, page 158)
Not documented.

VerticalAlignment(see TBaseVirtualTree.VerticalAlignment Property, page 158)
Used to set a node's vertical button aligment with regard to the entire node rectangle.

VisibleCount(see TBaseVirtualTree.VisibleCount Property, page 159)
Number of currently visible nodes.

VisiblePath(see TBaseVirtualTree.VisiblePath Property, page 159)
Property to set or determine a node parent's expand states.

WantTabs(see TBaseVirtualTree.WantTabs Property, page 159)
Read or set whether the tree wants to process tabs on its own.

Legend

published

Property

public

protected

read only

Event

Method

virtual

Class Hierarchy

File

VirtualTrees

10.1.15.1 TVirtualDrawTree.Action Property

Not documented.

Pascal

```
property Action;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.2 TVirtualDrawTree.Align Property

Not documented.

Pascal

```
property Align;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.3 TVirtualDrawTree.Alignment Property

Determines the horizontal alignment of text if no columns are defined.

Pascal

```
property Alignment: TAlignment;
```

Description

This property is only used if there are no columns defined and applies only to the node captions. Right alignment means here the right client area border and left aligned means the node buttons/lines etc. (both less the text margin).

Class

TVirtualDrawTree Class(see page 323)

10.1.15.4 TVirtualDrawTree.Anchors Property

Not documented.

Pascal

```
property Anchors;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.5 TVirtualDrawTree.AnimationDuration Property

Determines the maximum duration the tree can use to play an animation.

Pascal

```
property AnimationDuration: Cardinal;
```

Description

The value is specified in milliseconds and per default there are 200 ms as time frame, which is the recommended duration for such operations. On older systems (particularly Windows 95 and Windows 98) the animation process might not get enough CPU time to avoid expensive animations to finish properly. Still the animation loop tries to stay as close as possible to the given time.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.6 TVirtualDrawTree.AutoExpandDelay Property

Time delay after which a node gets expanded if it is the current drop target.

Pascal

```
property AutoExpandDelay: Cardinal;
```

Description

This value is specified in milliseconds and determines when to expand a node if it is the current drop target. This value is only used if voAutoDropExpand in Options is set.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.7 TVirtualDrawTree.AutoScrollDelay Property

Time which determines when auto scrolling should start.

Pascal

```
property AutoScrollDelay: Cardinal;
```

Description

Once the mouse pointer has been moved near to a border a timer is started using the interval specified by AutoScrollDelay. When the timer has fired auto scrolling starts provided it is enabled (see also TreeOptions). The value is specified in milliseconds.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.8 TVirtualDrawTree.AutoScrollInterval Property

Time interval between scroll events when doing auto scroll.

Pascal

```
property AutoScrollInterval: TAutoScrollInterval;
```

Description

This property determines the speed how the tree is scrolled vertically or horizontally when auto scrolling is in progress. The value is given in milliseconds.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.9 TVirtualDrawTree.Background Property

Holds a background image for the tree.

Pascal

```
property Background: TPicture;
```

Description

Virtual Treeview supports a fixed background image which does not scroll but can be adjusted by BackgroundOffsetX and BackgroundOffsetY.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.10 TVirtualDrawTree.BackgroundOffsetX Property

Horizontal offset of the background image.

Pascal

```
property BackgroundOffsetX: Integer;
```

Description

Determines the horizontal offset of the left border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

TVirtualDrawTree Class(see page 323)

10.1.15.11 TVirtualDrawTree.BackgroundOffsetY Property

Vertical offset of the background image.

Pascal

```
property BackgroundOffsetY: Integer;
```

Description

Determines the vertical offset of the top border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

TVirtualDrawTree Class(see page 323)

10.1.15.12 TVirtualDrawTree.BevelEdges Property

Not documented.

Pascal

```
property BevelEdges;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.13 TVirtualDrawTree.BevelInner Property

Not documented.

Pascal

```
property BevelInner;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.14 TVirtualDrawTree.BevelKind Property

Not documented.

Pascal

```
property BevelKind;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.15 TVirtualDrawTree.BevelOuter Property

Not documented.

Pascal

```
property BevelOuter;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.16 TVirtualDrawTree.BevelWidth Property

Not documented.

Pascal

```
property BevelWidth;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.17 TVirtualDrawTree.BiDiMode Property

Not documented.

Pascal

```
property BiDiMode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.18 TVirtualDrawTree.BorderStyle Property

Same as TForm.BorderStyle.

Pascal

```
property BorderStyle: TBorderStyle;
```

Description

See TForm.BorderStyle.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.19 TVirtualDrawTree.BorderWidth Property

Not documented.

Pascal

```
property BorderWidth;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.20 TVirtualDrawTree.ButtonFillMode Property

Determines how to fill the background of the node buttons.

Pascal

```
property ButtonFillMode: TVTButtonFillMode;
```

Description

This property is used to specify how the interior of the little plus and minus node buttons should be drawn, if ButtonStyle is bsTriangle.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.21 TVirtualDrawTree.ButtonStyle Property

Determines the look of node buttons.

Pascal

```
property ButtonStyle: TVTButtonStyle;
```

Description

Determines the look of node buttons.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.22 TVirtualDrawTree.Canvas Property

Not documented.

Pascal

```
property Canvas;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.23 TVirtualDrawTree.ChangeDelay Property

Time which determines when the OnChange event should be triggered after the actual change event.

Pascal

```
property ChangeDelay: Cardinal;
```

Description

In order to accumulate many quick changes in the tree you can use this delay value to specify after which wait time the OnChange event should occur. A value of 0 means to trigger OnChange immediately after the change (usually a selection or focus change) happend. Any value > 0 will start a timer which then triggers OnChange.

Note that there is the synchronous mode (started by BeginSynch) which effectively circumvents the change delay for the duration of the synchronous mode (stopped by EndSynch) regardless of the ChangeDelay setting.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.24 TVirtualDrawTree.CheckImageKind Property

Determines which images should be used for checkboxes and radio buttons.

Pascal

```
property CheckImageKind: TCheckImageKind;
```

Description

CheckImageKind can be used to switch the image set, which should be used for the tree. Read the description about TCheckImageKind for a list of all images, which can be used. CheckImageKind can also be set to ckCustom, which allows to supply a customized set of images to the tree. In order to have that working you must assign an image list (TCustomImageList) to the CustomCheckImages property.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.25 TVirtualDrawTree.ClipboardFormats Property

Special class to keep a list of clipboard format descriptions.

Pascal

```
property ClipboardFormats: TClipboardFormats;
```

Description

This TStringList descendant is used to keep a number of clipboard format descriptions, which are usually used to register clipboard formats with the system. Using a string list for this task allows to store enabled clipboard formats in the DFM.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.26 TVirtualDrawTree.Color Property

Not documented.

Pascal

```
property Color;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.27 TVirtualDrawTree.Colors Property

A collection of colors used in the tree.

Pascal

```
property Colors: TVTColors;
```

Description

This property holds an instance of the TVTColors class, which is used to customize many of the colors used in a tree. Placing them all in a specialized class helps organizing the colors in the object inspector and improves general management.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.28 TVirtualDrawTree.Constraints Property

Not documented.

Pascal

```
property Constraints;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.29 TVirtualDrawTree.Ctl3D Property

Not documented.

Pascal

```
property Ctl3D;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.30 TVirtualDrawTree.CustomCheckImages Property

Assign your own image list to get the check images you like most.

Pascal

```
property CustomCheckImages: TCustomImageList;
```

Description

The CustomCheckImages property is used when custom check images are enabled (see also ckCustom in TCheckImageKind).

See Also

TCheckImageKind

Class

TVirtualDrawTree Class(see page 323)

10.1.15.31 TVirtualDrawTree.DefaultNodeHeight Property

Read or set the height new nodes get as initial value.

Pascal

```
property DefaultNodeHeight: Cardinal;
```

Description

This property allows to read the current initial height for new nodes and to set a new value. Note that changing the

property value does not change the height of existing nodes. Only new nodes are affected.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.32 TVirtualDrawTree.DefaultPasteMode Property

Read or set the value, which determines where to add pasted nodes to.

Pascal

```
property DefaultPasteMode: TVTNodeAttachMode;
```

Description

The default paste mode is an attach mode, which is used when pasting data from the clipboard into the tree. Usually, you will want new nodes to be added as child nodes to the currently focused node (and this is also the default value), but you can also specify to add nodes only as siblings.

See Also

TVTNodeAttachMode

Class

TVirtualDrawTree Class(see page 323)

10.1.15.33 TVirtualDrawTree.DragCursor Property

Not documented.

Pascal

```
property DragCursor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.34 TVirtualDrawTree.DragHeight Property

Read or set the vertical limit of the internal drag image.

Pascal

```
property DragHeight: Integer;
```

Description

The DragHeight property (as well as the DragWidth property) are only for compatibility reason in the tree. If a platform does not support the IDropTargetHelper interface (Windows 9x/Me, Windows NT 4.0) then Virtual Treeview uses its own implementation of a DragImage. Since displaying a translucent drag image is performance hungry you should limit the image size shown for the drag operation.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.35 TVirtualDrawTree.DragImageKind Property

Read or set what should be shown in the drag image.

Pascal

```
property DragImageKind: TVTDragImageKind;
```

Description

DragImageKind allows to switch parts of the drag image off and on.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.36 TVirtualDrawTree.DragKind Property

Not documented.

Pascal

```
property DragKind;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.37 TVirtualDrawTree.DragMode Property

Not documented.

Pascal

```
property DragMode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.38 TVirtualDrawTree.DragOperations Property

Read or set which drag operations may be allowed in the tree.

Pascal

```
property DragOperations: TDragOperations;
```

Description

Using this property you can determine, which actions may be performed when a drag operation is finished. The default value includes move, copy and link, where link is rather an esoteric value and only there because it is supported by OLE. The values used directly determine which image is shown for the drag cursor. The specified drag operations do not tell which actions will actually be performed but only, which actions are allowed. They still can be modified during drag'n

drop by using a modifier key like the control, shift or alt key or can entirely be ignored by the drop handler.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.39 TVirtualDrawTree.DragType Property

Read or set which subsystem should be used for dragging.

Pascal

```
property DragType: TVTDragType;
```

Description

Traditionally, Delphi only supports its own drag mechanism, which is not compatible with the rest of the system. This VCL dragging also does not support to transport random data nor does it support drag operations between applications. Thus Virtual Treeview also supports the generally used OLE dragging, which in turn is incompatible with VCL dragging. Depending on your needs you can enable either VCL or OLE dragging as both together cannot be started. However, Virtual Treeview is able to act as drop target for both kind of data, independant of what is set in DragType.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.40 TVirtualDrawTree.DragWidth Property

Read or set the horizontal limit of the internal drag image.

Pascal

```
property DragWidth: Integer;
```

Description

The DragWidth property (as well as the DragHeight property) are only for compatibility reason in the tree. If a platform does not support the IDropTargetHelper interface (Windows 9x/Me, Windows NT 4.0) then Virtual Treeview uses its own implementation of a DragImage. Since displaying a translucent drag image is performance hungry you should limit the image size shown for the drag operation.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.41 TVirtualDrawTree.DrawSelectionMode Property

Read or set how multiselection with the mouse is to be visualized.

Pascal

```
property DrawSelectionMode: TVTDrawSelectionMode;
```

Description

Virtuall Treeview allows to display two different selection rectangles when doing multiselection with the mouse. One is the traditiional dotted focus rectangle and the other one is a translucent color rectangle. The latter is the preferred one but the former is set as default (for compatibility reasons).

Class

TVirtualDrawTree Class(see page 323)

10.1.15.42 TVirtualDrawTree.EditDelay Property

Read or set the maximum time between two single clicks on the same node, which should start node editing.

Pascal

```
property EditDelay: Cardinal;
```

Description

A node edit operation can be started using the keyboard (F2 key), in code using EditNode or by clicking twice on the same node (but not doing a double click). EditDelay is the maxmimum time distance between both clicks in which the edit operation is started.

See Also

Editors and editing(see page 41)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.43 TVirtualDrawTree.Enabled Property

Not documented.

Pascal

```
property Enabled;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.44 TVirtualDrawTree.Font Property

Same as TWinControl.Font.

Pascal

```
property Font;
```

Description

See TWinControl.Font.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.45 TVirtualDrawTree.Header Property

Provides access to the header instance.

Pascal

```
property Header: TVTHeader;
```

Description

This property is used to allow access to the header instance, which manages all aspects of the tree's header image as well as the column settings.

See Also

TVTHeader

Class

TVirtualDrawTree Class(see page 323)

10.1.15.46 TVirtualDrawTree.HintAnimation Property

Read or set the current hint animation type.

Pascal

```
property HintAnimation: THintAnimationType;
```

Description

With this property you can specify what animation you would like to play when displaying a hint. For some applications it might not be good to animate hints, hence you can entirely switch them off. Usually however you will leave the system standard. This way the user can decide whether and which hint animation he or she likes.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.47 TVirtualDrawTree.HintMode Property

Read or set what type of hint you want for the tree view.

Pascal

```
property HintMode: TVTHintMode;
```

Description

Virtual Treeview supports several hints modes. This includes the normal hint used for any other TControl class as well as a node specific hint, which is individual for each node or even each cell.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.48 TVirtualDrawTree.HotCursor Property

Read or set which cursor should be used for hot nodes.

Pascal

```
property HotCursor: TCursor;
```

Description

When you enable toHotTrack in TreeOptions.PaintOptions then the node, which is currently under the mouse pointer becomes the hot node. This is a special state, which can be used for certain effects. Hot nodes have by default an underlined caption and may cause the cursor to change to what ever you like. The HotCursor property is used to specify, which cursor is to be used.

See Also

HotNode, TVTPaintOptions

Class

TVirtualDrawTree Class(see page 323)

10.1.15.49 TVirtualDrawTree.Images Property

Read or set the tree's normal image list.

Pascal

```
property Images: TCustomImageList;
```

Description

Just like with TListView and TTreeview also Virtual Treeview can take an image list for its normal images. Additionally, there are image lists for state images and check images.

See Also

StateImages, CheckImages

Class

TVirtualDrawTree Class(see page 323)

10.1.15.50 TVirtualDrawTree.IncrementalSearch Property

Read or set the current incremental search mode.

Pascal

```
property IncrementalSearch: TVTIncrementalSearch;
```

Description

Virtual Treeview can do an incremental search by calling back the application when comparing node captions. The IncrementalSearch property determines whether incremental search is enabled and which nodes should be searched through.

See Also

IncrementalSearchDirection, IncrementalSearchStart, IncrementalSearchTimeout

Class

TVirtualDrawTree Class(see page 323)

10.1.15.51 TVirtualDrawTree.IncrementalSearchDirection Property

Read or set the direction to be used for incremental search.

Pascal

```
property IncrementalSearchDirection: TVTSearchDirection;
```

Description

When incremental search is enabled then Virtual Treeview can search forward and backward from the start point given by

IncrementalSearchStart.

See Also

IncrementalSearch, IncrementalSearchStart, IncrementalSearchTime123out

Class

TVirtualDrawTree Class(see page 323)

10.1.15.52 TVirtualDrawTree.IncrementalSearchStart Property

Read or set where to start incremental search.

Pascal

```
property IncrementalSearchStart: TVTSearchStart;
```

Description

When incremental search is enabled in the tree view then you can specify here, where to start the next incremental search operation from.

See Also

IncrementalSearch, IncrementalSearchDirection, IncrementalSearchTimeout

Class

TVirtualDrawTree Class(see page 323)

10.1.15.53 TVirtualDrawTree.IncrementalSearchTimeout Property

Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Pascal

```
property IncrementalSearchTimeout: Cardinal;
```

Description

When incremental search is enabled in Virtual Treeview then you can specify here after what time incremental search should stop when no keyboard input is encountered any longer. This property so determines also the speed at which users have to type letters to keep the incremental search rolling.

See Also

IncrementalSearch, IncrementalSearchDirection, IncrementalSearchStart

Class

TVirtualDrawTree Class(see page 323)

10.1.15.54 TVirtualDrawTree.Indent Property

Read or set the indentation amount for node levels.

Pascal

```
property Indent: Cardinal;
```

Description

Each new level in the tree (child nodes of a parent node) are visually shifted to distinguish betwenn them and their

parent node (that's the tree layout after all). The Indent property determines the shift distance in pixels.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.55 TVirtualDrawTree.LineMode Property

Read or set the mode of the tree lines.

Pascal

```
property LineMode: TVTLineMode;
```

Description

Apart from the usual lines Virtual Treeview also supports a special draw mode named bands. This allows for neat visual effects.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.56 TVirtualDrawTree.LineStyle Property

Read or set the mode of the tree lines.

Pascal

```
property LineStyle: TVTLineStyle;
```

Description

Virtual Treeview allows to customize the lines used to display the node hierarchy. The default style is a dotted pattern, but you can also make solid lines or specify your own line pattern.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.57 TVirtualDrawTree.Margin Property

Read or set the tree's node margin.

Pascal

```
property Margin: Integer;
```

Description

The node margin is the distance between the cell bounds and its content like the lines, images, check box and so on. However this border is only applied to the left and right side of the node cell.

Note: there is also a TextMargin property in TVirtualStringTree, which is an additional border for the cell text only.

See Also

TVirtualStringTree.TextMargin

Class

TVirtualDrawTree Class(see page 323)

10.1.15.58 TVirtualDrawTree.NodeAlignment Property

Read or set the node alignment value.

Pascal

```
property NodeAlignment: TVTNodeAlignment;
```

Description

Nodes have got an align member, which is used to determine the vertical position of the node's images and tree lines. The NodeAlignment property specifies how to interpret the value in the align member.

See Also

TVirtualNode

Class

TVirtualDrawTree Class(see page 323)

10.1.15.59 TVirtualDrawTree.NodeDataSize Property

Read or set the extra data size for each node.

Pascal

```
property NodeDataSize: Integer;
```

Description

A node can have an area for user data, which can be used to store application defined, node specific data in. Use GetNodeData to get the address of this area. In addition to assigning a value here you can also use the OnGetNodeDataSize event, which is called when NodeDataSize is -1.

See Also

Data handling(see page 39)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.60 TVirtualDrawTree.OnAdvancedHeaderDraw Event

Header paint support event.

Pascal

```
property OnAdvancedHeaderDraw: TVTAdvancedHeaderPaintEvent;
```

Description

The OnAdvancedHeaderDraw event is used when owner draw is enabled for the header and a column is set to owner draw mode. It can be used to custom draw only certain parts of the header instead the whole thing. A good example for this event is customizing the background of the header for only one column. With the standard custom draw method (OnHeaderDraw) you are in an all-or-nothing situation and have to paint everything in the header including the text, images and sort direction indicator. OnAdvancedHeaderDraw however uses OnHeaderDrawQueryElements to ask for the elements the application wants to draw and acts accordingly.

See Also

OnHeaderDrawQueryElements, OnHeaderDraw

Class

TVirtualDrawTree Class(see page 323)

10.1.15.61 TVirtualDrawTree.OnAfterCellPaint Event

Paint support event.

Pascal

```
property OnAfterCellPaint: TVTAfterCellPaintEvent;
```

Description

This event is called whenever a cell has been painted. A cell is defined as being one part of a node bound to a certain column. This event is called several times per node (the amount is determined by visible columns and size of the part to draw).

See Also

Paint cycles and stages(see page 36)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.62 TVirtualDrawTree.OnAfterItemErase Event

Paint support event.

Pascal

```
property OnAfterItemErase: TVTAfterItemEraseEvent;
```

Description

Called after the background of a node has been erased (erasing can also be filling with a background image). This event is called once per node in a paint cycle.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.63 TVirtualDrawTree.OnAfterItemPaint Event

Paint support event.

Pascal

```
property OnAfterItemPaint: TVTAfterItemPaintEvent;
```

Description

Called after a node has been drawn. This event is called once per node.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.64 TVirtualDrawTree.OnAfterPaint Event

Paint support event.

Pascal

```
property OnAfterPaint: TVTPaintEvent;
```

Description

Called after all nodes which needed an update have been drawn. This event is called once per paint cycle.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.65 TVirtualDrawTree.OnBeforeCellPaint Event

Paint support event.

Pascal

```
property OnBeforeCellPaint: TVTBeforeCellPaintEvent;
```

Description

This event is called immediately before a cell is painted.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.66 TVirtualDrawTree.OnBeforeItemErase Event

Paint support event.

Pascal

```
property OnBeforeItemErase: TVTBeforeItemEraseEvent;
```

Description

Called when the background of a node is about to be erased.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.67 TVirtualDrawTree.OnBeforeItemPaint Event

Paint support event.

Pascal

```
property OnBeforeItemPaint: TVTBeforeItemPaintEvent;
```

Description

Called after the background of a node has been drawn and just before the node itself is painted. In this event the application gets the opportunity to decide whether a node should be drawn normally or should be skipped. The application can draw the node itself if necessary or leave the node area blank.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.68 TVirtualDrawTree.OnBeforePaint Event

Paint support event.

Pascal

```
property OnBeforePaint: TVTPaintEvent;
```

Description

Called as very first event in a paint cycle. In this event has the application the opportunity to do some special preparation of the canvas onto which the tree is painted, e.g. setting a special viewport and origin or a different mapping mode.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.69 TVirtualDrawTree.OnChange Event

Navigation support event.

Pascal

```
property OnChange: TVTChangeEvent;
```

Description

Called when a node's selection state has changed.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.70 TVirtualDrawTree.OnChecked Event

Check support event.

Pascal

```
property OnChecked: TVTChangeEvent;
```

Description

Triggered when a node's check state has changed.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.71 TVirtualDrawTree.OnChecking Event

Check support event.

Pascal

```
property OnChecking: TVTCheckChangingEvent;
```

Description

Triggered when a node's check state is about to change and allows to prevent the change.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.72 TVirtualDrawTree.OnClick Property

Not documented.

Pascal

```
property OnClick;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.73 TVirtualDrawTree.OnCollapsed Event

Miscellaneous event.

Pascal

```
property OnCollapsed: TVTChangeEvent;
```

Description

Triggered after a node has been collapsed, that is, its child nodes are no longer displayed.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.74 TVirtualDrawTree.OnCollapsing Event

Miscellaneous event.

Pascal

```
property OnCollapsing: TVTChangingEvent;
```

Description

Triggered when a node is about to be collapsed and allows to prevent collapsing the node by setting Allowed to false.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.75 TVirtualDrawTree.OnColumnClick Event

Header and column support event.

Pascal

```
property OnColumnClick: TVTColumnClickEvent;
```

Description

Triggered when the user released a mouse button over the same column in the client area on which the button was pressed previously.

See Also

OnHeaderClick

Class

TVirtualDrawTree Class(see page 323)

10.1.15.76 TVirtualDrawTree.OnColumnDblClick Event

Header and column support event.

Pascal

```
property OnColumnDblClick: TVTColumnDblClickEvent;
```

Description

Same as OnColumnClick but for double clicks.

See Also

OnColumnClick, OnHeaderDblClick

Class

TVirtualDrawTree Class(see page 323)

10.1.15.77 TVirtualDrawTree.OnColumnResize Event

Header and column support routine.

Pascal

```
property OnColumnResize: TVTHeaderNotifyEvent;
```

Description

Triggered when a column is being resized. During resize OnColumnResize is frequently hence you should make any code in the associated event handle a short and fast as possible.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.78 TVirtualDrawTree.OnCompareNodes Event

Sort and search support event.

Pascal

```
property OnCompareNodes: TVTCompareEvent;
```

Description

This event is the core event for all comparations between nodes. It is important that you write a handler for this event if you want to sort nodes!

Result must be set to less than 0 if Node1 is considered as being before Node2, equal to 0 if both a considered being the same and greater than 0 if the first node is considered as being after node 2. Keep in mind that you don't need to take sort direction into account. This is automatically handled by the tree. Simply return a comparation result as would there be an ascending sort order.

Below is some sample code taken from the Advanced Demo:

```
procedure TMainForm.VDT1CompareNodes(Sender: TBaseVirtualTree; Node1, Node2: PVirtualNode; Column:
Integer;
  var Result: Integer);

// used to sort the image draw tree

var
  Data1,
  Data2: PImageData;

begin
  Data1 := Sender.GetNodeData(Node1);
  Data2 := Sender.GetNodeData(Node2);
  // folder are always before files
  if Data1.IsFolder <> Data2.IsFolder then
  begin
    // one of both is a folder the other a file
    if Data1.IsFolder then
      Result := -1
    else
      Result := 1;
  end
  else // both are of same type (folder or file)
    Result := CompareText(Data1.FullPath, Data2.FullPath);
end;
```

See Also

SortTree, Sort

Class

TVirtualDrawTree Class(see page 323)

10.1.15.79 TVirtualDrawTree.OnCreateDataObject Event

Drag'n drop support event.

Pascal

```
property OnCreateDataObject: TVTCreateDataObjectEvent;
```

Description

This event is called when the tree's drag manager needs a data object interface to start a drag'n drop operation. Descentants (which override DoGetDataObject) or the application can return an own IDataObject implementation to support special formats.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.80 TVirtualDrawTree.OnCreateDragManager Event

Drag'n drop support event.

Pascal

```
property OnCreateDragManager: TVTCreateDragManagerEvent;
```

Description

This event is usually not used but allows power users to create their own drag manager to have different actions and/or formats than the internal drag manager.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.81 TVirtualDrawTree.OnCreateEditor Event

Editing support event.

Pascal

```
property OnCreateEditor: TVTCreateEditorEvent;
```

Description

Allows to supply a customized node editor without changing the tree. TBaseVirtualTree triggers this event and raises an exception if there no editor is returned. If you don't want this then disable edit support for nodes in TreeOptions.MiscOptions. Descentants like TCustomVirtualStringTree supply a generic and simple string editor.

See Also

Editors and editing(see page 41)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.82 TVirtualDrawTree.OnDblClick Property

Not documented.

Pascal

```
property OnDblClick;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.83 TVirtualDrawTree.OnDragAllowed Event

Drag'n drop support event.

Pascal

```
property OnDragAllowed: TVTDragAllowedEvent;
```

Description

This event is called in the mouse button down handler to determine whether the application allows to start a drag operation. Since this check is done in sync with the other code it is much prefered over doing a manual BeginDrag.

Notes

The OnDragAllowed event is called only if the current DragMode is dmManual.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.84 TVirtualDrawTree.OnDragDrop Event

Drag'n drop support event.

Pascal

```
property OnDragDrop: TVTDragDropEvent;
```

Description

Triggered when either a VCL or a OLE drop action occured. Accepting drag and drop actions is not trivial. In order to maintain a minimum compatibility with the VCL drag'n drop system Virtual Tree accepts not only OLE drop actions but also those issued by the Delphi VCL (which is totally different to the OLE way, unfortunately), provided toAcceptOLEDrop is set in TreeOptions.MiscOptions. The code snippet below is taken from a sample project provided with Virtual Tree. It shows a general way to deal with dropped data. The following check list can be used as orientation and additional comment to the code:

1. Determine what kind of drop data is passed. If DataObject is nil or Formats is empty then the drag source is a VCL control. The event is not triggered for OLE drag'n drop if there is no OLE format is available (which should never occur).
2. If the event is triggered by a VCL control then use Source to access either the control or the drag object, depending on the circumstances of the action.
3. For OLE drag'n drop iterate through the Formats list to find a format you can handle.
4. If you find CF_VIRTUALTREE then the source of the drag operation is a Virtual Treeview. Since this is the native tree format you can pass it to the Sender's ProcessDrop method which will take care to retrieve the data and act depending on Effect and Mode. No further action by the application is usually required in this case.
5. If you do not find CF_VIRTUALTREE then the operation has been initiated by another application, e.g. the Explorer (then you will find CF_HDROP or CF_SHELLIDLIST in formats) or Notepad (then you will get CF_TEXT and perhaps CF_UNICODETEXT) etc., depending on the data which is actually dropped.
6. Use the provided DataObject to get the drop data via IDataObject.GetData and act depending on the format you get.
7. Finally set Effect to either DROPEFFECT_COPY, DROPEFFECT_MOVE or DROPEFFECT_NONE to indicate which operation

needs to be finished in Sender when the event returns. If you return DROPEFFECT_MOVE then all marked nodes in the source tree will be deleted, otherwise they stay where they are.

```
procedure TMainForm.VTDragDrop(Sender: TBaseVirtualTree; Source: TObject; DataObject: IDataObject;
  const Formats: array of Word; Shift: TShiftState; Pt: TPoint; var Effect: Integer; Mode: TDropMode);

var
  I: Integer;
  AttachMode: TVTNodeAttachMode;

begin
  if Length(Formats) > 0 then
  begin
    // OLE drag'n drop
    // If the native tree format is listed then use this and accept the drop, otherwise recject
(ignore) it.
    // It is recommend by Microsoft to order available clipboard formats in decreasing detail richness
so
    // the first best format which we can accept is usually the best format we can get at all.
    for I := 0 to High(Formats) do
      if Formats[I] = CF_VIRTUALTREE then
      begin
        case Mode of
          dmAbove:
            AttachMode := amInsertBefore;
          dmOnNode:
            AttachMode := amAddChildLast;
          dmBelow:
            AttachMode := amInsertAfter;
        else
          if Assigned(Source) and (Source is TBaseVirtualTree) and (Sender <> Source) then
            AttachMode := amInsertBefore
          else
            AttachMode := amNowhere;
        end;
        // in the case the drop target does an optimized move Effect is set to DROPEFFECT_NONE
        // to indicate this also to the drag source (so the source doesn't need to take any further
action)
        Sender.ProcessDrop(DataObject, Sender.DropTargetNode, Effect, AttachMode);
        Break;
      end;
  end
  else
  begin
    // VCL drag'n drop, Effects contains by default both move and copy effect suggestion,
    // as usual the application has to find out what operation is finally to do
    Beep;
  end;
end;
```

Class

TVirtualDrawTree Class(see page 323)

10.1.15.85 TVirtualDrawTree.OnDragOver Event

Drag'n drop support event.

Pascal

```
property OnDragOver: TVTDragOverEvent;
```

Description

Triggered when Sender is the potential target of a drag'n drop operation. You can use this event to allow or deny a drop operation by setting Allowed to True or False, respectively. For conditions of OLE or VCL drag source see OnDragDrop.

See Also

OnDragDrop

Class

TVirtualDrawTree Class(see page 323)

10.1.15.86 TVirtualDrawTree.OnDrawHint Event

Triggered when a node hint or tooltip must be drawn.

Pascal

```
property OnDrawHint: TVTDrawHintEvent;
```

Description

Use an event handler for OnDrawHint to draw the hint or tooltip for the given node. You must implement this event and OnGetHintSize to get a hint at all.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.87 TVirtualDrawTree.OnDrawNode Event

Triggered when a node must be drawn.

Pascal

```
property OnDrawNode: TVTDrawNodeEvent;
```

Description

Use an event handler for OnDrawNode to draw the actual content for the given node.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.88 TVirtualDrawTree.OnEdited Event

Editing support event.

Pascal

```
property OnEdited: TVTEditChangeEvent;
```

Description

Triggered when an edit action has successfully been finished.

See Also

Editors and editing(see page 41)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.89 TVirtualDrawTree.OnEditing Event

Editing support event.

Pascal

```
property OnEditing: TVTEditChangingEvent;
```

Description

Triggered when a node is about to be edited. Use Allowed to allow or deny this action.

See Also

Editors and editing(see page 41)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.90 TVirtualDrawTree.OnEndDock Property

Not documented.

Pascal

```
property OnEndDock;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.91 TVirtualDrawTree.OnEndDrag Property

Not documented.

Pascal

```
property OnEndDrag;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.92 TVirtualDrawTree.OnEnter Property

Not documented.

Pascal

```
property OnEnter;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.93 TVirtualDrawTree.OnExit Property

Not documented.

Pascal

```
property OnExit;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.94 TVirtualDrawTree.OnExpanded Event

Misscellaneous event.

Pascal

```
property OnExpanded: TVTChangeEvent;
```

Description

Triggered after a node has been expanded.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.95 TVirtualDrawTree.OnExpanding Event

Miscellaneous event.

Pascal

```
property OnExpanding: TVTChangingEvent;
```

Description

Triggered just before a node is expanded. Use Allowed to allow or deny this action.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.96 TVirtualDrawTree.OnFocusChanged Event

Navigation support event.

Pascal

```
property OnFocusChanged: TVTFocusChangeEvent;
```

Description

Triggered after the focused node changed. When examining Node keep in mind that it can be nil, meaning there is no

focused node.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.97 TVirtualDrawTree.OnFocusChanging Event

Navigation support event.

Pascal

```
property OnFocusChanging: TVTFocusChangingEvent;
```

Description

Triggered when the node focus is about to change. You can use Allowed to allow or deny a focus change. Keep in mind that either the old or the new node can be nil.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.98 TVirtualDrawTree.OnFreeNode Event

Data management node.

Pascal

```
property OnFreeNode: TVTFreeNodeEvent;
```

Description

Triggered when a node is about to be freed. This is the ideal place to free/disconnect your own data you associated with Node. Keep in mind, that data which is stored directly in the node does not need to be free by the application. This is part of the node record and will be freed when the node is freed. You should however finalize the data in such a case if it contains references to external memory objects (e.g. variants, strings, interfaces).

Class

TVirtualDrawTree Class(see page 323)

10.1.15.99 TVirtualDrawTree.OnGetCellIsEmpty Event

Triggered when the tree control needs to know whether a given column is empty.

Pascal

```
property OnGetCellIsEmpty: TVTGetCellIsEmptyEvent;
```

Description

Virtual Treeview supports the concept of column spanning where one cell with too much text to fit into its own space can expand to the right cell neighbors if they are empty. To make this work it is necessary to know if a cell is considered as being empty, whatever this means to an application. The string tree descendant simply checks the text for the given cell and calls back its ancestor if there is no text to further refine if the cell must stay as if it contained something. The ancestor (TBaseVirtualTree) now triggers OnGetCellIsEmpty to let the application decide.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.100 TVirtualDrawTree.OnGetCursor Event

Miscellaneous event.

Pascal

```
property OnGetCursor: TVTGetCursorEvent;
```

Description

This event is triggered from the WM_SETCURSOR message to allow the application use several individual cursors for a tree. The Cursor property allows to set one cursor for the whole control but not to use separate cursors for different tree parts.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.101 TVirtualDrawTree.OnGetHeaderCursor Event

Header and column support event.

Pascal

```
property OnGetHeaderCursor: TVTGetHeaderCursorEvent;
```

Description

This event is triggered from the WM_SETCURSOR message to allow the application to define individual cursors for the header part of the tree control.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.102 TVirtualDrawTree.OnGetHelpContext Event

Miscellaneous event.

Pascal

```
property OnGetHelpContext: TVTHelpContextEvent;
```

Description

This event is usually triggered when the user pressed F1 while the tree has the focus. The tree is iteratively traversed all the way up to the top level parent of the given node until a valid help context index is returned (via this event). When the loop reaches the top level without getting a help index then the tree control's help index is used. If the tree itself does not have a help context index then a further traversal is initiated going up parent by parent of each control in the current window hierarchy until either a valid index is found or there is no more window parent.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.103 TVirtualDrawTree.OnGetHintSize Event

Triggered when a node hint or tooltip is about to show.

Pascal

```
property OnGetHintSize: TVTGetHintSizeEvent;
```

Description

Use an event handler for OnGetHintSize to return the size of the tooltip/hint window for the given node. You must implement this event and OnDrawHint to get a hint at all.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.104 TVirtualDrawTree.OnGetImageIndex Event

Display management event.

Pascal

```
property OnGetImageIndex: TVTGetImageEvent;
```

Description

This event is triggered whenever the tree needs the index of an image, be it the normal, the selected or the state image. The event should be as fast as possible because it is at times frequently called when the layout of the node must be determined, e.g. while doing draw selection with the mouse or painting the tree. Kind determines which image is needed and Column determines for which column of the node the image is needed. This value can be -1 to indicate there is no column used. The parameter Ghosted can be set to true to blend the image 50% against the tree background and can be used for instance in explorer trees to mark hidden file system objects. Additionally nodes are also drawn with a ghosted icon if the are part of a cut set during a pending cut-to-clipboard operation. In this case changing the ghosted parameter has no effect.

Notes

Blending nodes can be switched by using toUseBlendImages in TreeOptions.PaintOptions.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.105 TVirtualDrawTree.OnGetImageIndexEx Event

Not documented.

Pascal

```
property OnGetImageIndexEx: TVTGetImageExEvent;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.106 TVirtualDrawTree.OnGetLineStyle Event

Display management event.

Pascal

```
property OnGetLineStyle: TVTGetLineStyleEvent;
```

Description

This event is used to customize the appearance of the tree and grid lines and is only triggered if the LineStyle property is set to lsCustomStyle. The event must return a pointer to an array containing bits for an 8 x 8 pixel image with word aligned entries. For more info see PrepareBitmaps and the Windows APIs CreateBitmap and CreatePatternBrush.

Notes

It is important that you do not use dynamically allocated memory in this event (also no local variables on the stack). If you do so then either the memory is not valid on return of the event (if allocated on stack) or will never be freed (if allocated with a memory manager). Instead use a constant array and return its address.

See Also

PrepareBitmaps

Class

TVirtualDrawTree Class(see page 323)

10.1.15.107 TVirtualDrawTree.OnGetNodeDataSize Event

Data management event.

Pascal

```
property OnGetNodeDataSize: TVTGetNodeDataSizeEvent;
```

Description

Triggered when access to a node's data happens the first time but the actual data size is not yet set. Usually you would specify the size of the data you want to have added to each node by NodeDataSize, e.g. SizeOf(TMyRecord) is quite usual there (where TMyRecord is the structure you want to have stored in the node). Sometimes, however it is not possible to determine the node size in advance, so you can leave NodeDataSize being -1 (the default value) and the OnGetNodeDataSize event is triggered as soon as the first regular node is created (the hidden root node does not have user data but internal data which is determined by other means).

See Also

NodeDataSize, Data handling(see page 39)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.108 TVirtualDrawTree.OnGetNodeWidth Event

Triggered when a node is about to be drawn.

Pascal

```
property OnGetNodeWidth: TVTGetNodeWidthEvent;
```

Description

Use an event handler for OnGetNodeWidth to return your calculated width for the given node. Since the draw does not know the width of a node you have to tell it yourself.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.109 TVirtualDrawTree.OnGetPopupMenu Event

Miscellaneous event.

Pascal

```
property OnGetPopupMenu: TVTPopupEvent;
```

Description

This event allows the application to return a popup menu which is specific to a certain node. The tree does an automatic traversal all the way up to the top level node which is the parent of a given node to get a popup menu. If Menu is set then the traversal stops. Otherwise it continues until either a menu is set, AskParent is set to False or the top level parent has been reached.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.110 TVirtualDrawTree.OnGetUserClipboardFormats Event

Drag'n drop and clipboard support event.

Pascal

```
property OnGetUserClipboardFormats: TVTGetUserClipboardFormatsEvent;
```

Description

Whenever the tree needs to specify the available clipboard formats for a clipboard or drag'n drop operation it calls this event too, to allow the application or descentants (which would override DoGetUserClipboardFormats) to specify own formats which can be rendered. Since the build-in data object does not know how to render formats which are specified here you have to supply a handler for the OnRenderOLEData event or an own IDataObject implementation to fully support your own formats.

Use the Formats parameter which is an open array and add the identifiers of your formats (which you got when you registered the format).

Class

TVirtualDrawTree Class(see page 323)

10.1.15.111 TVirtualDrawTree.OnHeaderClick Event

Header & column support event.

Pascal

```
property OnHeaderClick: TVTHeaderClickEvent;
```

Description

This event is triggered when the user clicks on a header button and is usually a good place to set the current SortColumn and SortDirection.

See Also

SortColumn, SortDirection

Class

TVirtualDrawTree Class(see page 323)

10.1.15.112 TVirtualDrawTree.OnHeaderDblClick Event

Header & column support event.

Pascal

```
property OnHeaderDblClick: TVTHeaderClickEvent;
```

Description

Unlike OnHeaderClick this event is triggered for double clicks on any part of the header and comes with more detailed information like shift state, which mouse button caused the event and the mouse position.

See Also

OnHeaderClick

Class

TVirtualDrawTree Class(see page 323)

10.1.15.113 TVirtualDrawTree.OnHeaderDragged Event

Header & column support event.

Pascal

```
property OnHeaderDragged: TVTHeaderDraggedEvent;
```

Description

Triggered after the user has released the left mouse button when a header drag operation was active. Column contains the index of the column which was dragged. Use this index for the Columns property of the header to find out the current position. OldPosition is the position which Column occupied before it was dragged around.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.114 TVirtualDrawTree.OnHeaderDraggedOut Event

Header & column support event.

Pascal

```
property OnHeaderDraggedOut: TVTHeaderDraggedOutEvent;
```

Description

When during a header drag operation the mouse moves out of the header rectangle and the mouse button is released then an OnHeaderDraggedOut event will be fired with the target mouse position in screen coordinates.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.115 TVirtualDrawTree.OnHeaderDragging Event

Header & column support event.

Pascal

```
property OnHeaderDragging: TVTHeaderDraggingEvent;
```

Description

Triggered just before dragging of a header button starts. Set Allowed to False if you want to prevent the drag operation of the given column.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.116 TVirtualDrawTree.OnHeaderDraw Event

Header & column support event.

Pascal

```
property OnHeaderDraw: TVTHeaderPaintEvent;
```

Description

If you set the hoOwnerDraw style in TVTHeader.Options and a column has been set to vsOwnerDraw (see also TVirtualTreeColumn.Style) then OnDrawHeader is called whenever a column needs painting.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.117 TVirtualDrawTree.OnHeaderDrawQueryElements Event

Header & column support event.

Pascal

```
property OnHeaderDrawQueryElements: TVTHeaderPaintQueryElementsEvent;
```

Description

Used for advanced header painting to query the application for the elements, which are drawn by it and which should be drawn by the tree.

See Also

OnAdvancedHeaderDraw

Class

TVirtualDrawTree Class(see page 323)

10.1.15.118 TVirtualDrawTree.OnHeaderMouseDown Event

Header & column support event.

Pascal

```
property OnHeaderMouseDown: TVTHeaderMouseEvent;
```

Description

This event is similar to OnHeaderClick but comes with more detailed information like shift state, which mouse button caused the event and the mouse position.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.119 TVirtualDrawTree.OnHeaderMouseMove Event

Header & column support event.

Pascal

```
property OnHeaderMouseMove: TVTHeaderMouseMoveEvent;
```

Description

This event is triggered when the mouse pointer is moved over the header area.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.120 TVirtualDrawTree.OnHeaderMouseUp Event

Header & column support event.

Pascal

```
property OnHeaderMouseUp: TVTHeaderMouseEvent;
```

Description

This event is very much like OnHeaderMouseDown but is triggered when a mouse button is released.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.121 TVirtualDrawTree.OnHotChange Event

Navigation support event.

Pascal

```
property OnHotChange: TVTHotNodeChangeEvent;
```

Description

This event is triggered if hot tracking is enabled (see also TreeOptions.PaintOptions) and when the mouse pointer moves from one node caption to another. In full row select mode most parts of a node are considered as being part of the caption.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.122 TVirtualDrawTree.OnIncrementalSearch Event

Miscellaneous event.

Pascal

```
property OnIncrementalSearch: TVTIncrementalSearchEvent;
```

Description

This event is integral part of the incremental search functionality (see also Keyboard, hotkeys and incremental search). It is triggered during search for a node which matches the given string. Similar to other compare routines return a value < 0 if the node's caption is considered as being before the given text, = 0 if it is the same and > 0 if it is considered being after the given text.

```
procedure TfrmProperties.VST3IncrementalSearch(Sender: TBaseVirtualTree; Node: PVirtualNode; const
Text: WideString;
  var Result: Integer);

var
  S, PropText: string;


begin
  // Note: This code requires a proper Unicode/WideString comparison routine which I did not want to
link here for
  // size and clarity reasons. For now strings are (implicitly) converted to ANSI to make the
comparison work.
  // Search is not case sensitive.
  S := Text;
  if Node.Parent = Sender.RootNode then
  begin
    // root nodes
    if Node.Index = 0 then
      PropText := 'Description'
    else
      PropText := 'Origin';
  end
  else
  begin
    PropText := PropertyTexts[Node.Parent.Index, Node.Index, ptkText];
  end;

  // By using StrLIComp we can specify a maximum length to compare. This allows us to find also nodes
  // which match only partially.
  Result := StrLIComp(PChar(S), PChar(PropText), Min(Length(S), Length(PropText)))
end;
```

Notes

Usually incremental search allows to match also partially. Hence it is recommended to do comparison only up to the length

of the shorter string.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.123 TVirtualDrawTree.OnInitChildren Event

Node management event.

Pascal

```
property OnInitChildren: TVTInitChildrenEvent;
```

Description

In order to allow the tree only to fill content where needed it is possible to set the vsHasChildren style in a node's initializaton whithout really adding any child nodes. These child nodes must be initialized first when they are about to be displayed or another access (like search, iteration etc.) occurs.

The application usually prepares data needed to fill child nodes when they are initialized and retrieves the actual number. Set ChildCount to the number of children you want.

See Also

The virtual paradigm(see page 33)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.124 TVirtualDrawTree.OnInitNode Event

Node management event.

Pascal

```
property OnInitNode: TVTInitNodeEvent;
```

Description

This event is important to connect the tree to your internal data. It is the ideal place to put references or whatever you need into a node's data area. You can set some initial states like selection, expansion state or that a node has child nodes.

See Also

The virtual paradigm(see page 33)

Class

TVirtualDrawTree Class(see page 323)

10.1.15.125 TVirtualDrawTree.OnKeyAction Event

Miscellaneous event.

Pascal

```
property OnKeyAction: TVTKeyActionEvent;
```

Description

This event is a convinient way for the application or descentant trees to change the semantic of a certain key stroke. It is triggered when the user presses a key and allows either to process that key normally (leave DoDefault being True) or change it to another key instead (set DoDefault to False then). This way a key press can change its meaning or entirely be ignored (if CharCode is set to 0).

Class

TVirtualDrawTree Class(see page 323)

10.1.15.126 TVirtualDrawTree.OnKeyDown Property

Not documented.

Pascal

```
property OnKeyDown;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.127 TVirtualDrawTree.OnKeyPress Property

Not documented.

Pascal

```
property OnKeyPress;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.128 TVirtualDrawTree.OnKeyUp Property

Not documented.

Pascal

```
property OnKeyUp;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.129 TVirtualDrawTree.OnLoadNode Event

Streaming support event.

Pascal

```
property OnLoadNode: TVTSaveNodeEvent;
```

Description

This event is typically triggered when serialized tree data must be restored, e.g. when loading the tree from file or stream or during a clipboard/drag'n drop operation. You should only read in what you wrote out in OnSaveNode. For safety there is a check in the loader code which tries to keep the internal serialization structure intact in case the application does not read correctly.

See Also

OnSaveNode, LoadFromStream, SaveToStream, AddFromStream, VTTreeStreamVersion, TVTHeader.LoadFromStream, TVTHeader.SaveToStream

Class

TVirtualDrawTree Class(see page 323)

10.1.15.130 TVirtualDrawTree.OnMeasureItem Event

Miscellaneous event.

Pascal

```
property OnMeasureItem: TVTMeasureItemEvent;
```

Description

Virtual Treeview supports individual node heights. However it might sometimes unpractical to set this height in advance (e.g. during OnInitNode). Another scenario might be that multi line nodes must size themselves to accomodate the entire node text without clipping. For such and similar cases the event OnMeasureItem is for. It is queried once for each node and allows to specify the node's future height. If you later want to have a new height applied (e.g. because the node's text changed) then call InvalidateNode for it and its vsHeightMeasured state is reset causing so the tree to trigger the OnMeasureItem event again when the node is painted the next time.

See Also

InvalidateNode, vsHeightMeasured

Class

TVirtualDrawTree Class(see page 323)

10.1.15.131 TVirtualDrawTree.OnMouseDown Property

Not documented.

Pascal

```
property OnMouseDown;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.132 TVirtualDrawTree.OnMouseMove Property

Not documented.

Pascal

```
property OnMouseMove;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.133 TVirtualDrawTree.OnMouseUp Property

Not documented.

Pascal

```
property OnMouseUp;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.134 TVirtualDrawTree.OnMouseWheel Property

Not documented.

Pascal

```
property OnMouseWheel;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.135 TVirtualDrawTree.OnNodeCopied Event

Miscellaneous event.

Pascal

```
property OnNodeCopied: TVTNodeCopiedEvent;
```

Description

This event is triggered during drag'n drop after a node has been copied to a new location. Sender is the target tree where the copy operation took place.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.136 TVirtualDrawTree.OnNodeCopying Event

Miscellaneous event.

Pascal

```
property OnNodeCopying: TVTNodeCopyingEvent;
```

Description

This event is triggered when a node is about to be copied to a new location. Use Allowed to allow or deny the action. Sender is the target tree where the copy operation will take place.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.137 TVirtualDrawTree.OnNodeMoved Event

Miscellaneous event.

Pascal

```
property OnNodeMoved: TVTNodeMovedEvent;
```

Description

This event is very much like OnNodeCopied but used for moving nodes instead.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.138 TVirtualDrawTree.OnNodeMoving Event

Miscellaneous event.

Pascal

```
property OnNodeMoving: TVTNodeMovingEvent;
```

Description

This event is very much like OnNodeCopying but used for moving nodes instead.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.139 TVirtualDrawTree.OnPaintBackground Event

Paint support event.

Pascal

```
property OnPaintBackground: TVTBackgroundPaintEvent;
```

Description

This event is triggered when the tree has finished its painting and there is an area which is not covered by nodes. For nodes there are various events to allow background customizaton. For the free area in the tree window there is this event.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.140 TVirtualDrawTree.OnRenderOLEData Event

Drag'n drop and clipboard support event.

Pascal

```
property OnRenderOLEData: TVTRenderOLEDataEvent;
```

Description

This event is triggered when the data in a clipboard or drag'n drop operation must be rendered but the built-in data object does not know the requested format. This is usually the case when the application (or descentants) have specified their own formats in OnGetUserClipboardFormats.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.141 TVirtualDrawTree.OnResetNode Event

Node management event.

Pascal

```
property OnResetNode: TVTChangeEvent;
```

Description

For large trees or simply because the content changed it is sometimes necessary to discard a certain node and release all its children. This can be done with ResetNode which will trigger this event.

See Also

ResetNode

Class

TVirtualDrawTree Class(see page 323)

10.1.15.142 TVirtualDrawTree.OnResize Property

Not documented.

Pascal

```
property OnResize;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.143 TVirtualDrawTree.OnSaveNode Event

Streaming support event.

Pascal

```
property OnSaveNode: TVTSaveNodeEvent;
```

Description

This event is triggered whenever a certain node must be serialized into a stream, e.g. for saving to file or for copying to another tree/node during a clipboard or drag'n drop operation. Make sure you only store non-transient data into the stream. Pointers (including long/wide string references) are transient and the application cannot assume to find the data a pointer references on saving at the same place when the node is loaded (see also OnLoadNode). This is even more essential for nodes which are moved or copied between different trees in different processes (applications). Storing strings however is easily done by writing the strings as a whole into the stream.

Notes

For exchanging data between different trees and for general stability improvement I strongly recommend that you insert a kind of identifier as first stream entry when saving a node. This identifier can then be used to determine what data will follow when loading the node later and does normally not required to be stored in the node data.

See Also

OnLoadNode, LoadFromStream, SaveToStream, AddFromStream, VTTreeStreamVersion, TVTHeader.LoadFromStream, TVTHeader.SaveToStream

Class

TVirtualDrawTree Class(see page 323)

10.1.15.144 TVirtualDrawTree.OnScroll Event

Miscellaneous event.

Pascal

```
property OnScroll: TVTScrollEvent;
```

Description

This event is triggered when the tree is scrolled horizontally or vertically. You can use it to synchronize scrolling of several trees or other controls.

See Also

OffsetXY

Class

TVirtualDrawTree Class(see page 323)

10.1.15.145 TVirtualDrawTree.OnShowScrollbar Event

Not documented.

Pascal

```
property OnShowScrollbar: TVTScrollbarShowEvent;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.146 TVirtualDrawTree.OnStartDock Property

Not documented.

Pascal

```
property OnStartDock;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.147 TVirtualDrawTree.OnStateChange Event

Miscellaneous event.

Pascal

```
property OnStateChange: TVTStateChangeEvent;
```

Description

For special effects or in order to increase performance it is sometimes useful to know when the tree changes one of its internal states like tsIncrementalSearching or tsOLEDragging. The OnStateChange event is triggered each time such a change occurs letting so the application take measures for it.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.148 TVirtualDrawTree.OnStructureChange Event

Miscellaneous event.

Pascal

```
property OnStructureChange: TVTStructureChangeEvent;
```

Description

This event is triggered when a change in the tree structure is made. That means whenever a node is created or destroyed or a node's child list is change (because a child node was moved, copied etc.) then OnStructureChange is executed.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.149 TVirtualDrawTree.OnUpdating Event

Miscellaneous event.

Pascal

```
property OnUpdating: TVTUpdatingEvent;
```

Description

This event is triggered when the application or the tree call BeginUpdate or EndUpdate and indicate so when a larger update operation takes place. This can for instance be used to show a hour glass wait cursor.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.150 TVirtualDrawTree.ParentBiDiMode Property

Not documented.

Pascal

```
property ParentBiDiMode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.151 TVirtualDrawTree.ParentColor Property

Not documented.

Pascal

```
property ParentColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.152 TVirtualDrawTree.ParentCtl3D Property

Not documented.

Pascal

```
property ParentCtl3D;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.153 TVirtualDrawTree.ParentFont Property

Not documented.

Pascal

```
property ParentFont;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.154 TVirtualDrawTree.ParentShowHint Property

Not documented.

Pascal

```
property ParentShowHint;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.155 TVirtualDrawTree.PopupMenu Property

Not documented.

Pascal

```
property PopupMenu;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.156 TVirtualDrawTree.RootNodeCount Property

Read or set the number of nodes on the top level.

Pascal

```
property RootNodeCount: Cardinal;
```

Description

Usually setting RootNodeCount is all what is needed to initially fill the tree. When one of the top level nodes is initialized you can set its ivsHasChildren style. This will then cause to ask to initialize the child nodes. Recursively applied, you can use this principle to create tree nodes on demand (e.g. when their parent is expanded).

Class

TVirtualDrawTree Class(see page 323)

10.1.15.157 TVirtualDrawTree.ScrollBarOptions Property

Reference to the scroll bar options class.

Pascal

```
property ScrollBarOptions: TScrollBarOptions;
```

Description

Like many other aspects in Virtual Treeview also scrollbars can be customized. See the class itself for further descriptions.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.158 TVirtualDrawTree.SelectionBlendFactor Property

Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

Pascal

```
property SelectionBlendFactor: Byte;
```

Description

For a visually appealing tree some operations use alpha blending. One of these operations is multi selection using the mouse. Another one is the rectangle drawn around the caption of selected nodes. Both rectangles use the SelectionBlendFactor to determine how much of the underlying tree image and how much of the rectangles should be seen. The factor can be in the range of [0..255] where 0 means the rectangle is fully transparent and 255 it is fully opaque.

If you don't like to use blended node selection rectangles then switch them off by removing toUseBlendedSelection from TVTPaintOptions. For selecting a certain multi selection rectangle style use DrawSelectionMode.

Notes

Alpha blending is only enabled when the current processor supports MMX instructions. If MMX is not supported then a dotted draw selection rectangle and an opaque node selection rectangle is used.

See Also

DrawSelectionMode, TVTPaintOptions

Class

TVirtualDrawTree Class(see page 323)

10.1.15.159 TVirtualDrawTree.SelectionCurveRadius Property

Read or set the current corner radius for node selection rectangles.

Pascal

```
property SelectionCurveRadius: Cardinal;
```

Description

This is a special property to determine the radius of the corners of the selection rectangle for a node caption. Virtual Treeview supports not only simple rectangular selection marks but also such with rounded corners. This feature, however, is only available if blended node selection rectangles are disabled.

See Also

SelectionBlendFactor, DrawSelectionMode, TVTPaintOptions

Class

TVirtualDrawTree Class(see page 323)

10.1.15.160 TVirtualDrawTree.ShowHint Property

Not documented.

Pascal

```
property ShowHint;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.161 TVirtualDrawTree.StateImages Property

Reference to the images list which is used for the state images.

Pascal

```
property StateImages: TCustomImageList;
```

Description

Each node can (in each column) have several images. One is the check image which is supplied by internal image lists or a special external list (see also CustomCheckImages). Another one is the state image and yet another one the normal/selected image.

See Also

CheckImages, Images

Class

TVirtualDrawTree Class(see page 323)

10.1.15.162 TVirtualDrawTree.TabOrder Property

Not documented.

Pascal

```
property TabOrder;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.163 TVirtualDrawTree.TabStop Property

Not documented.

Pascal

```
property TabStop;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.164 TVirtualDrawTree.TextMargin Property

Read or set the distance of the node caption to its borders.

Pascal

```
property TextMargin: Integer;
```

Description

TextMargin is used to define a border like area within the content rectangle of a node. This rectangle is the area of the node less the space used for indentation, images, lines and node margins and usually contains the text of a node. In order to support finer adjustment there is another margin, which only applies to the left and right border in the content rectangle. This is the text margin.

See Also

Margin

Class

TVirtualDrawTree Class(see page 323)

10.1.15.165 TVirtualDrawTree.TreeOptions Property

Reference to the tree's options.

Pascal

```
property TreeOptions: TVirtualTreeOptions;
```

Description

The tree options are one of the main switchs to modify a treeview's behavior. Virtual Treeview supports customizing tree options by descentants. This allows very fine adjustments for derived tree classes, including the decision which properties should be published. For more information about the base options see TCustomVirtualTreeOptions and its descentants.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.166 TVirtualDrawTree.Visible Property

Not documented.

Pascal

```
property Visible;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.167 TVirtualDrawTree.WantTabs Property

Read or set whether the tree wants to process tabs on its own.

Pascal

```
property WantTabs: Boolean;
```

Description

Usually tab kex strokes advance the input focus from one control to another on a form. For special processing however it is necessary to let the control decide what to do with the given tabulator character. Virtual Treeview needs this character mainly for its grid emulation.

Class

TVirtualDrawTree Class(see page 323)

10.1.15.168 TVirtualDrawTree.GetOptionsClass Method

Customization helper to determine which options class the tree should use.

Pascal

```
function GetOptionsClass: TTreeOptionsClass; override;
```

Description

GetOptionsClass is a special purpose method to return a certain class which is used by the tree for its options. TVirtualBaseTree always returns TCustomVirtualTreeOptions but descendants can override this method to return own classes.

For ease of use it makes much sense to always use the same name for the tree's options (which is TreeOptions). By using a customized options class, however, the wrong type is returned by this property. Hence it is meaningful to override TreeOptions and return the derived options class. To make this work the tree descendant must additionally provide new access methods for this property. An example can be seen in TVirtualStringTree:

```
  TVirtualStringTree = class(TCustomVirtualStringTree)
  private
    function GetOptions: TStringTreeOptions;
    procedure SetOptions(const Value: TStringTreeOptions);
  protected
    function GetOptionsClass: TTreeOptionsClass; override;
  public
    property Canvas;
  published
    ...
    property TreeOptions: TStringTreeOptions read GetOptions write SetOptions;
    ...
  end;

  ...

//----------------- TVirtualStringTree
--------------------------------------------------------------------------------

function TVirtualStringTree.GetOptions: TStringTreeOptions;

begin
  Result := FOptions as TStringTreeOptions;
end;

//----------------------------------------------------------------------------------------------------------
-----------------

procedure TVirtualStringTree.SetOptions(const Value: TStringTreeOptions);

begin
  FOptions.Assign(Value);
```

```
end;

//----------------------------------------------------------------------------------------------------
-----------------

function TVirtualStringTree.GetOptionsClass: TTreeOptionsClass;

begin
  Result := TStringTreeOptions;
end;
```

Class

TVirtualDrawTree Class(see page 323)

10.1.16 TVirtualStringTree Class

Descentant of TBaseVirtualTree(see TBaseVirtualTree Class, page 88) which is able to manage node captions on its own.

Pascal

```
TVirtualStringTree = class(TCustomVirtualStringTree);
```

Description

TVirtualStringTree adds no new functionality to TCustomVirtualStringTree(see TCustomVirtualStringTree Class, page 274) but is publicly available version and appears in the component palette.

Events

OnAdvancedHeaderDraw(see TVirtualStringTree.OnAdvancedHeaderDraw Event, page 447)
Header paint support event.

OnAfterCellPaint(see TVirtualStringTree.OnAfterCellPaint Event, page 448)
Paint support event.

OnAfterItemErase(see TVirtualStringTree.OnAfterItemErase Event, page 448)
Paint support event.

OnAfterItemPaint(see TVirtualStringTree.OnAfterItemPaint Event, page 448)
Paint support event.

OnAfterPaint(see TVirtualStringTree.OnAfterPaint Event, page 449)
Paint support event.

OnBeforeCellPaint(see TVirtualStringTree.OnBeforeCellPaint Event, page 449)
Paint support event.

OnBeforeItemErase(see TVirtualStringTree.OnBeforeItemErase Event, page 449)
Paint support event.

OnBeforeItemPaint(see TVirtualStringTree.OnBeforeItemPaint Event, page 449)
Paint support event.

OnBeforePaint(see TVirtualStringTree.OnBeforePaint Event, page 450)
Paint support event.

OnChange(see TVirtualStringTree.OnChange Event, page 450)
Navigation support event.

OnChecked(see TVirtualStringTree.OnChecked Event, page 450)
Check support event.

OnChecking(see TVirtualStringTree.OnChecking Event, page 451)
Check support event.

OnCollapsed(see TVirtualStringTree.OnCollapsed Event, page 451)
Miscellaneous event.

OnCollapsing(see TVirtualStringTree.OnCollapsing Event, page 451)
Miscellaneous event.

OnColumnClick(see TVirtualStringTree.OnColumnClick Event, page 452)
Header and column support event.

OnColumnDblClick(see TVirtualStringTree.OnColumnDblClick Event, page 452)
Header and column support event.

OnColumnResize(see TVirtualStringTree.OnColumnResize Event, page 452)
Header and column support routine.

OnCompareNodes(see TVirtualStringTree.OnCompareNodes Event, page 453)
Sort and search support event.

OnCreateDataObject(see TVirtualStringTree.OnCreateDataObject Event, page 453)
Drag'n drop support event.

OnCreateDragManager(see TVirtualStringTree.OnCreateDragManager Event, page 454)
Drag'n drop support event.

OnCreateEditor(see TVirtualStringTree.OnCreateEditor Event, page 454)
Editing support event.

OnDragAllowed(see TVirtualStringTree.OnDragAllowed Event, page 455)
Drag'n drop support event.

OnDragDrop(see TVirtualStringTree.OnDragDrop Event, page 455)
Drag'n drop support event.

OnDragOver(see TVirtualStringTree.OnDragOver Event, page 456)
Drag'n drop support event.

OnEditCancelled(see TVirtualStringTree.OnEditCancelled Event, page 457)
Editing support event.

OnEdited(see TVirtualStringTree.OnEdited Event, page 457)
Editing support event.

OnEditing(see TVirtualStringTree.OnEditing Event, page 457)
Editing support event.

OnExpanded(see TVirtualStringTree.OnExpanded Event, page 458)
Misscellaneous event.

OnExpanding(see TVirtualStringTree.OnExpanding Event, page 459)
Miscellaneous event.

OnFocusChanged(see TVirtualStringTree.OnFocusChanged Event, page 459)
Navigation support event.

OnFocusChanging(see TVirtualStringTree.OnFocusChanging Event, page 459)
Navigation support event.

OnFreeNode(see TVirtualStringTree.OnFreeNode Event, page 460)
Data management node.

OnGetCellIsEmpty(see TVirtualStringTree.OnGetCellIsEmpty Event, page 460)
Triggered when the tree control needs to know whether a given column is empty.

OnGetCursor(see TVirtualStringTree.OnGetCursor Event, page 460)
Miscellaneous event.

OnGetHeaderCursor(see TVirtualStringTree.OnGetHeaderCursor Event, page 460)
Header and column support event.

OnGetHelpContext(see TVirtualStringTree.OnGetHelpContext Event, page 461)
Miscellaneous event.

OnGetHint(see TVirtualStringTree.OnGetHint Event, page 461)
Virtual string tree event to query for a custom hint text.

OnGetImageIndex(see TVirtualStringTree.OnGetImageIndex Event, page 461)
Display management event.

OnGetImageIndexEx(see TVirtualStringTree.OnGetImageIndexEx Event, page 462)

Not documented.

OnGetLineStyle(see TVirtualStringTree.OnGetLineStyle Event, page 462)
Display management event.

OnGetNodeDataSize(see TVirtualStringTree.OnGetNodeDataSize Event, page 463)
Data management event.

OnGetPopupMenu(see TVirtualStringTree.OnGetPopupMenu Event, page 463)
Miscellaneous event.

OnGetText(see TVirtualStringTree.OnGetText Event, page 463)
Virtual string tree event to query for a node's normal or static text.

OnGetUserClipboardFormats(see TVirtualStringTree.OnGetUserClipboardFormats Event, page 464)
Drag'n drop and clipboard support event.

OnHeaderClick(see TVirtualStringTree.OnHeaderClick Event, page 464)
Header & column support event.

OnHeaderDblClick(see TVirtualStringTree.OnHeaderDblClick Event, page 465)
Header & column support event.

OnHeaderDragged(see TVirtualStringTree.OnHeaderDragged Event, page 465)
Header & column support event.

OnHeaderDraggedOut(see TVirtualStringTree.OnHeaderDraggedOut Event, page 465)
Header & column support event.

OnHeaderDragging(see TVirtualStringTree.OnHeaderDragging Event, page 466)
Header & column support event.

OnHeaderDraw(see TVirtualStringTree.OnHeaderDraw Event, page 466)
Header & column support event.

OnHeaderDrawQueryElements(see TVirtualStringTree.OnHeaderDrawQueryElements Event, page 466)
Header & column support event.

OnHeaderMouseDown(see TVirtualStringTree.OnHeaderMouseDown Event, page 467)
Header & column support event.

OnHeaderMouseMove(see TVirtualStringTree.OnHeaderMouseMove Event, page 467)
Header & column support event.

OnHeaderMouseUp(see TVirtualStringTree.OnHeaderMouseUp Event, page 467)
Header & column support event.

OnHotChange(see TVirtualStringTree.OnHotChange Event, page 467)
Navigation support event.

OnIncrementalSearch(see TVirtualStringTree.OnIncrementalSearch Event, page 468)
Miscellaneous event.

OnInitChildren(see TVirtualStringTree.OnInitChildren Event, page 469)
Node management event.

OnInitNode(see TVirtualStringTree.OnInitNode Event, page 469)
Node management event.

OnKeyAction(see TVirtualStringTree.OnKeyAction Event, page 469)
Miscellaneous event.

OnLoadNode(see TVirtualStringTree.OnLoadNode Event, page 470)
Streaming support event.

OnMeasureItem(see TVirtualStringTree.OnMeasureItem Event, page 471)
Miscellaneous event.

OnNewText(see TVirtualStringTree.OnNewText Event, page 472)
Virtual string tree event to pass edited text.

OnNodeCopied(see TVirtualStringTree.OnNodeCopied Event, page 473)
Miscellaneous event.

OnNodeCopying(see TVirtualStringTree.OnNodeCopying Event, page 473)
Miscellaneous event.

OnNodeMoved(see TVirtualStringTree.OnNodeMoved Event, page 473)
Miscellaneous event.

OnNodeMoving(see TVirtualStringTree.OnNodeMoving Event, page 473)
Miscellaneous event.

OnPaintBackground(see TVirtualStringTree.OnPaintBackground Event, page 474)
Paint support event.

OnPaintText(see TVirtualStringTree.OnPaintText Event, page 474)
Event to change text formatting for particular nodes.

OnRenderOLEData(see TVirtualStringTree.OnRenderOLEData Event, page 475)
Drag'n drop and clipboard support event.

OnResetNode(see TVirtualStringTree.OnResetNode Event, page 475)
Node management event.

OnSaveNode(see TVirtualStringTree.OnSaveNode Event, page 475)
Streaming support event.

OnScroll(see TVirtualStringTree.OnScroll Event, page 476)
Miscellaneous event.

OnShortenString(see TVirtualStringTree.OnShortenString Event, page 476)
String tree event for custom handling of string abbreviations.

OnShowScrollbar(see TVirtualStringTree.OnShowScrollbar Event, page 477)
Not documented.

OnStateChange(see TVirtualStringTree.OnStateChange Event, page 477)
Miscellaneous event.

OnStructureChange(see TVirtualStringTree.OnStructureChange Event, page 478)
Miscellaneous event.

OnUpdating(see TVirtualStringTree.OnUpdating Event, page 478)
Miscellaneous event.

TCustomVirtualStringTree Class

OnGetHint(see TCustomVirtualStringTree.OnGetHint Event, page 295)
Virtual string tree event to query for a custom hint text.

OnGetText(see TCustomVirtualStringTree.OnGetText Event, page 295)
Virtual string tree event to query for a node's normal or static text.

OnNewText(see TCustomVirtualStringTree.OnNewText Event, page 296)
Virtual string tree event to pass edited text.

OnPaintText(see TCustomVirtualStringTree.OnPaintText Event, page 297)
Event to change text formatting for particular nodes.

OnShortenString(see TCustomVirtualStringTree.OnShortenString Event, page 297)
String tree event for custom handling of string abbreviations.

TBaseVirtualTree Class

OnAdvancedHeaderDraw(see TBaseVirtualTree.OnAdvancedHeaderDraw Event, page 128)
Header paint support event.

OnAfterCellPaint(see TBaseVirtualTree.OnAfterCellPaint Event, page 128)
Paint support event.

OnAfterItemErase(see TBaseVirtualTree.OnAfterItemErase Event, page 129)
Paint support event.

OnAfterItemPaint(see TBaseVirtualTree.OnAfterItemPaint Event, page 129)
Paint support event.

OnAfterPaint(see TBaseVirtualTree.OnAfterPaint Event, page 129)
Paint support event.

OnBeforeCellPaint(see TBaseVirtualTree.OnBeforeCellPaint Event, page 130)
Paint support event.

OnBeforeItemErase(see TBaseVirtualTree.OnBeforeItemErase Event, page 130)
Paint support event.

OnBeforeItemPaint(see TBaseVirtualTree.OnBeforeItemPaint Event, page 130)
Paint support event.

OnBeforePaint(see TBaseVirtualTree.OnBeforePaint Event, page 131)
Paint support event.

OnChange(see TBaseVirtualTree.OnChange Event, page 131)
Navigation support event.

OnChecked(see TBaseVirtualTree.OnChecked Event, page 131)
Check support event.

OnChecking(see TBaseVirtualTree.OnChecking Event, page 131)
Check support event.

OnCollapsed(see TBaseVirtualTree.OnCollapsed Event, page 132)
Miscellaneous event.

OnCollapsing(see TBaseVirtualTree.OnCollapsing Event, page 132)
Miscellaneous event.

OnColumnClick(see TBaseVirtualTree.OnColumnClick Event, page 132)
Header and column support event.

OnColumnDblClick(see TBaseVirtualTree.OnColumnDblClick Event, page 132)
Header and column support event.

OnColumnResize(see TBaseVirtualTree.OnColumnResize Event, page 133)
Header and column support routine.

OnCompareNodes(see TBaseVirtualTree.OnCompareNodes Event, page 133)
Sort and search support event.

OnCreateDataObject(see TBaseVirtualTree.OnCreateDataObject Event, page 134)
Drag'n drop support event.

OnCreateDragManager(see TBaseVirtualTree.OnCreateDragManager Event, page 134)
Drag'n drop support event.

OnCreateEditor(see TBaseVirtualTree.OnCreateEditor Event, page 134)
Editing support event.

OnDragAllowed(see TBaseVirtualTree.OnDragAllowed Event, page 135)
Drag'n drop support event.

OnDragDrop(see TBaseVirtualTree.OnDragDrop Event, page 135)
Drag'n drop support event.

OnDragOver(see TBaseVirtualTree.OnDragOver Event, page 137)
Drag'n drop support event.

OnEditCancelled(see TBaseVirtualTree.OnEditCancelled Event, page 137)
Editing support event.

OnEdited(see TBaseVirtualTree.OnEdited Event, page 137)
Editing support event.

OnEditing(see TBaseVirtualTree.OnEditing Event, page 138)
Editing support event.

OnExpanded(see TBaseVirtualTree.OnExpanded Event, page 138)
Misscellaneous event.

OnExpanding(see TBaseVirtualTree.OnExpanding Event, page 138)
Miscellaneous event.

OnFocusChanged(see TBaseVirtualTree.OnFocusChanged Event, page 138)
Navigation support event.

OnFocusChanging(see TBaseVirtualTree.OnFocusChanging Event, page 139)
Navigation support event.

OnFreeNode(see TBaseVirtualTree.OnFreeNode Event, page 139)

Data management node.

OnGetCellIsEmpty(see TBaseVirtualTree.OnGetCellIsEmpty Event, page 139)
Triggered when the tree control needs to know whether a given column is empty.

OnGetCursor(see TBaseVirtualTree.OnGetCursor Event, page 140)
Miscellaneous event.

OnGetHeaderCursor(see TBaseVirtualTree.OnGetHeaderCursor Event, page 140)
Header and column support event.

OnGetHelpContext(see TBaseVirtualTree.OnGetHelpContext Event, page 140)
Miscellaneous event.

OnGetImageIndex(see TBaseVirtualTree.OnGetImageIndex Event, page 140)
Display management event.

OnGetImageIndexEx(see TBaseVirtualTree.OnGetImageIndexEx Event, page 141)
Not documented.

OnGetLineStyle(see TBaseVirtualTree.OnGetLineStyle Event, page 141)
Display management event.

OnGetNodeDataSize(see TBaseVirtualTree.OnGetNodeDataSize Event, page 142)
Data management event.

OnGetPopupMenu(see TBaseVirtualTree.OnGetPopupMenu Event, page 142)
Miscellaneous event.

OnGetUserClipboardFormats(see TBaseVirtualTree.OnGetUserClipboardFormats Event, page 142)
Drag'n drop and clipboard support event.

OnHeaderClick(see TBaseVirtualTree.OnHeaderClick Event, page 143)
Header & column support event.

OnHeaderDblClick(see TBaseVirtualTree.OnHeaderDblClick Event, page 143)
Header & column support event.

OnHeaderDragged(see TBaseVirtualTree.OnHeaderDragged Event, page 143)
Header & column support event.

OnHeaderDraggedOut(see TBaseVirtualTree.OnHeaderDraggedOut Event, page 144)
Header & column support event.

OnHeaderDragging(see TBaseVirtualTree.OnHeaderDragging Event, page 144)
Header & column support event.

OnHeaderDraw(see TBaseVirtualTree.OnHeaderDraw Event, page 144)
Header & column support event.

OnHeaderDrawQueryElements(see TBaseVirtualTree.OnHeaderDrawQueryElements Event, page 145)
Header & column support event.

OnHeaderMouseDown(see TBaseVirtualTree.OnHeaderMouseDown Event, page 145)
Header & column support event.

OnHeaderMouseMove(see TBaseVirtualTree.OnHeaderMouseMove Event, page 145)
Header & column support event.

OnHeaderMouseUp(see TBaseVirtualTree.OnHeaderMouseUp Event, page 145)
Header & column support event.

OnHotChange(see TBaseVirtualTree.OnHotChange Event, page 146)
Navigation support event.

OnIncrementalSearch(see TBaseVirtualTree.OnIncrementalSearch Event, page 146)
Miscellaneous event.

OnInitChildren(see TBaseVirtualTree.OnInitChildren Event, page 147)
Node management event.

OnInitNode(see TBaseVirtualTree.OnInitNode Event, page 147)
Node management event.

OnKeyAction(see TBaseVirtualTree.OnKeyAction Event, page 148)
Miscellaneous event.

OnLoadNode(see TBaseVirtualTree.OnLoadNode Event, page 148)
Streaming support event.

OnMeasureItem(see TBaseVirtualTree.OnMeasureItem Event, page 148)
Miscellaneous event.

OnNodeCopied(see TBaseVirtualTree.OnNodeCopied Event, page 149)
Miscellaneous event.

OnNodeCopying(see TBaseVirtualTree.OnNodeCopying Event, page 149)
Miscellaneous event.

OnNodeMoved(see TBaseVirtualTree.OnNodeMoved Event, page 149)
Miscellaneous event.

OnNodeMoving(see TBaseVirtualTree.OnNodeMoving Event, page 150)
Miscellaneous event.

OnPaintBackground(see TBaseVirtualTree.OnPaintBackground Event, page 150)
Paint support event.

OnRenderOLEData(see TBaseVirtualTree.OnRenderOLEData Event, page 150)
Drag'n drop and clipboard support event.

OnResetNode(see TBaseVirtualTree.OnResetNode Event, page 151)
Node management event.

OnSaveNode(see TBaseVirtualTree.OnSaveNode Event, page 151)
Streaming support event.

OnScroll(see TBaseVirtualTree.OnScroll Event, page 152)
Miscellaneous event.

OnShowScrollbar(see TBaseVirtualTree.OnShowScrollbar Event, page 152)
Not documented.

OnStateChange(see TBaseVirtualTree.OnStateChange Event, page 152)
Miscellaneous event.

OnStructureChange(see TBaseVirtualTree.OnStructureChange Event, page 152)
Miscellaneous event.

OnUpdating(see TBaseVirtualTree.OnUpdating Event, page 153)
Miscellaneous event.

Group

Classes(see page 86)

Methods

GetOptionsClass(see TVirtualStringTree.GetOptionsClass Method , page 484)
Customization helper to determine which options class the tree should use.

TCustomVirtualStringTree Class

AdjustPaintCellRect(see TCustomVirtualStringTree.AdjustPaintCellRect Method , page 298)
Method which can be used by descentants to adjust the given rectangle during a paint cycle.

CalculateTextWidth(see TCustomVirtualStringTree.CalculateTextWidth Method , page 299)
Not documented.

ColumnIsEmpty(see TCustomVirtualStringTree.ColumnIsEmpty Method , page 299)
Used to determine if a cell is considered as being empty.

ComputeNodeHeight(see TCustomVirtualStringTree.ComputeNodeHeight Method , page 299)
Not documented.

ContentToClipboard(see TCustomVirtualStringTree.ContentToClipboard Method , page 299)
Not documented.

ContentToHTML(see TCustomVirtualStringTree.ContentToClipboard Method , page 299)
Not documented.

ContentToRTF(see TCustomVirtualStringTree.ContentToClipboard Method , page 299)
Not documented.

ContentToText(see TCustomVirtualStringTree.ContentToClipboard Method , page 299)
Not documented.

ContentToUnicode(see TCustomVirtualStringTree.ContentToClipboard Method , page 299)
Not documented.

Create(see TCustomVirtualStringTree.Create Constructor , page 300)
Constructor of the control

DefineProperties(see TCustomVirtualStringTree.DefineProperties Method , page 300)
Helper method to customize loading and saving persistent tree data.

DoCreateEditor(see TCustomVirtualStringTree.DoCreateEditor Method , page 300)
Not documented.

DoGetNodeHint(see TCustomVirtualStringTree.DoGetNodeHint Method , page 301)
Not documented.

DoGetNodeTooltip(see TCustomVirtualStringTree.DoGetNodeTooltip Method , page 301)
Not documented.

DoGetNodeWidth(see TCustomVirtualStringTree.DoGetNodeWidth Method , page 301)
Overridable method which always retuns 0.

DoGetText(see TCustomVirtualStringTree.DoGetText Method , page 301)
Not documented.

DoIncrementalSearch(see TCustomVirtualStringTree.DoIncrementalSearch Method , page 302)
Not documented.

DoNewText(see TCustomVirtualStringTree.DoNewText Method , page 302)
Not documented.

DoPaintNode(see TCustomVirtualStringTree.DoPaintNode Method , page 302)
Overridable method which does nothing.

DoPaintText(see TCustomVirtualStringTree.DoPaintText Method , page 302)
Not documented.

DoShortenString(see TCustomVirtualStringTree.DoShortenString Method , page 303)
Not documented.

DoTextDrawing(see TCustomVirtualStringTree.DoTextDrawing Method , page 303)
Not documented.

DoTextMeasuring(see TCustomVirtualStringTree.DoTextMeasuring Method , page 303)
Not documented.

GetOptionsClass(see TCustomVirtualStringTree.GetOptionsClass Method , page 303)
Customization helper to determine which options class the tree should use.

GetTextInfo(see TCustomVirtualStringTree.GetTextInfo Method , page 304)
Helper method for node editors, hints etc.

InternalData(see TCustomVirtualStringTree.InternalData Method , page 305)
Returns the address of the internal data for a tree class.

InvalidateNode(see TCustomVirtualStringTree.InvalidateNode Method , page 305)
Invalidates the given node.

MainColumnChanged(see TCustomVirtualStringTree.MainColumnChanged Method , page 305)
Not documented.

Path(see TCustomVirtualStringTree.Path Method , page 306)
Not documented.

ReadChunk(see TCustomVirtualStringTree.ReadChunk Method , page 306)
Not documented.

ReadOldStringOptions(see TCustomVirtualStringTree.ReadOldStringOptions Method , page 306)
Not documented.

ReinitNode(see TCustomVirtualStringTree.ReinitNode Method , page 306)
Forces a reinitialization of the given node.

RenderOLEData(see TCustomVirtualStringTree.RenderOLEData Method , page 307)

Renders pending OLE data.

WriteChunks(see TCustomVirtualStringTree.WriteChunks Method , page 307)
Writes the core chunks for the given node to the given stream.

TBaseVirtualTree Class

AbsoluteIndex(see TBaseVirtualTree.AbsoluteIndex Method , page 160)
Reads the overall index of a node.

AddChild(see TBaseVirtualTree.AddChild Method , page 160)
Creates and adds a new child node to given node.

AddFromStream(see TBaseVirtualTree.AddFromStream Method , page 161)
Adds the content from the given stream to the given node.

AddToSelection(see TBaseVirtualTree.AddToSelection Method (PVirtualNode), page 161)
Adds one or more nodes to the current selection.

AdjustPaintCellRect(see TBaseVirtualTree.AdjustPaintCellRect Method , page 162)
Used in descentants to modify the clip rectangle of the current column while painting a certain node.

AdjustPanningCursor(see TBaseVirtualTree.AdjustPanningCursor Method , page 162)
Loads the proper cursor which indicates into which direction scrolling is done.

AdviseChangeEvent(see TBaseVirtualTree.AdviseChangeEvent Method , page 162)
Used to register a delayed change event.

AllocateInternalDataArea(see TBaseVirtualTree.AllocateInternalDataArea Method , page 163)
Registration method to allocate tree internal data per node.

Animate(see TBaseVirtualTree.Animate Method , page 163)
Support method for animated actions in the tree view.

Assign(see TBaseVirtualTree.Assign Method , page 164)
Used to copy properties from another Virtual Treeview.

BeginDrag(see TBaseVirtualTree.BeginDrag Method , page 164)
Starts an OLE drag'n drop operation.

BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164)
Enters the tree into a special synchronized mode.

BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164)
Locks the tree view to perform several update operations.

CalculateSelectionRect(see TBaseVirtualTree.CalculateSelectionRect Method , page 165)
Support method for draw selection.

CanAutoScroll(see TBaseVirtualTree.CanAutoScroll Method , page 165)
Determines whether the tree can currently auto scroll its window.

CancelCutOrCopy(see TBaseVirtualTree.CancelCutOrCopy Method , page 165)
Canceles any pending cut or copy clipboard operation.

CancelEditNode(see TBaseVirtualTree.CancelEditNode Method , page 166)
Cancel the current edit operation, if there is any.

CanEdit(see TBaseVirtualTree.CanEdit Method , page 166)
Determines whether a node can be edited or not.

CanFocus(see TBaseVirtualTree.CanFocus Method , page 166)
Support method to determine whether the tree window can receive the input focus.

CanShowDragImage(see TBaseVirtualTree.CanShowDragImage Method , page 166)
Determines whether a drag image should be shown.

Change(see TBaseVirtualTree.Change Method , page 167)
Central method called when a node's selection state changes.

ChangeScale(see TBaseVirtualTree.ChangeScale Method , page 167)
Helper method called by the VCL when control resizing is due.

CheckParentCheckState(see TBaseVirtualTree.CheckParentCheckState Method , page 167)
Helper method for recursive check state changes.

Clear(see TBaseVirtualTree.Clear Method , page 168)

Clears the tree and removes all nodes.

ClearChecked(see TBaseVirtualTree.ClearChecked Method , page 168)
Not documented.

ClearSelection(see TBaseVirtualTree.ClearSelection Method , page 168)
Removes all nodes from the current selection.

ClearTempCache(see TBaseVirtualTree.ClearTempCache Method , page 168)
Helper method to clear(see TBaseVirtualTree.Clear Method , page 168) the internal temporary node cache.

ColumnIsEmpty(see TBaseVirtualTree.ColumnIsEmpty Method , page 169)
Used to determine if a cell is considered as being empty.

CopyTo(see TBaseVirtualTree.CopyTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 169)
Copies Source and all its child nodes to Target.

CopyToClipBoard(see TBaseVirtualTree.CopyToClipBoard Method , page 169)
Copies all currently selected nodes to the clipboard.

CountLevelDifference(see TBaseVirtualTree.CountLevelDifference Method , page 170)
Determines the level difference of two nodes.

CountVisibleChildren(see TBaseVirtualTree.CountVisibleChildren Method , page 170)
Determines the number of visible child nodes of the given node.

Create(see TBaseVirtualTree.Create Constructor , page 170)
Constructor of the control

CreateParams(see TBaseVirtualTree.CreateParams Method , page 171)
Prepares the creation of the controls window handle.

CreateWnd(see TBaseVirtualTree.CreateWnd Method , page 171)
Initializes data, which depends on the window handle.

CutToClipBoard(see TBaseVirtualTree.CutToClipBoard Method , page 171)
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties(see TBaseVirtualTree.DefineProperties Method , page 171)
Helper method to customize loading and saving persistent tree data.

DeleteChildren(see TBaseVirtualTree.DeleteChildren Method , page 172)
Removes all child nodes from the given node.

DeleteNode(see TBaseVirtualTree.DeleteNode Method , page 172)
Removes the given node from the tree.

DeleteSelectedNodes(see TBaseVirtualTree.DeleteSelectedNodes Method , page 172)
Removes all currently selected nodes form the tree.

Destroy(see TBaseVirtualTree.Destroy Destructor , page 173)
Destructor of the control.

DetermineHiddenChildrenFlag(see TBaseVirtualTree.DetermineHiddenChildrenFlag Method , page 173)
Determines whether all children of a given node are hidden.

DetermineHiddenChildrenFlagAllNodes(see TBaseVirtualTree.DetermineHiddenChildrenFlagAllNodes Method , page 173)
Determines whether all children of all nodes are hidden.

DetermineHitPositionLTR(see TBaseVirtualTree.DetermineHitPositionLTR Method , page 174)
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineHitPositionRTL(see TBaseVirtualTree.DetermineHitPositionLTR Method , page 174)
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineNextCheckState(see TBaseVirtualTree.DetermineNextCheckState Method , page 174)
Not documented.

DetermineScrollDirections(see TBaseVirtualTree.DetermineScrollDirections Method , page 174)
Not documented.

DoAdvancedHeaderDraw(see TBaseVirtualTree.DoAdvancedHeaderDraw Method , page 174)
Not documented.

DoAfterCellPaint(see TBaseVirtualTree.DoAfterCellPaint Method , page 175)
Not documented.

DoAfterItemErase(see TBaseVirtualTree.DoAfterItemErase Method , page 175)
Not documented.

DoAfterItemPaint(see TBaseVirtualTree.DoAfterItemPaint Method , page 175)
Not documented.

DoAfterPaint(see TBaseVirtualTree.DoAfterPaint Method , page 175)
Not documented.

DoAutoScroll(see TBaseVirtualTree.DoAutoScroll Method , page 176)
Enables or disables the auto scroll timer.

DoBeforeCellPaint(see TBaseVirtualTree.DoBeforeCellPaint Method , page 176)
Not documented.

DoBeforeDrag(see TBaseVirtualTree.DoBeforeDrag Method , page 176)
Not documented.

DoBeforeItemErase(see TBaseVirtualTree.DoBeforeItemErase Method , page 177)
Not documented.

DoBeforeItemPaint(see TBaseVirtualTree.DoBeforeItemPaint Method , page 177)
Not documented.

DoBeforePaint(see TBaseVirtualTree.DoBeforePaint Method , page 177)
Not documented.

DoCancelEdit(see TBaseVirtualTree.DoCancelEdit Method , page 177)
Called when the tree should stop editing without accepting changed values.

DoCanEdit(see TBaseVirtualTree.DoCanEdit Method , page 178)
Not documented.

DoChange(see TBaseVirtualTree.DoChange Method , page 178)
Not documented.

DoCheckClick(see TBaseVirtualTree.DoCheckClick Method , page 178)
Not documented.

DoChecked(see TBaseVirtualTree.DoChecked Method , page 178)
Not documented.

DoChecking(see TBaseVirtualTree.DoChecking Method , page 179)
Not documented.

DoCollapsed(see TBaseVirtualTree.DoCollapsed Method , page 179)
Not documented.

DoCollapsing(see TBaseVirtualTree.DoCollapsing Method , page 179)
Not documented.

DoColumnClick(see TBaseVirtualTree.DoColumnClick Method , page 179)
Not documented.

DoColumnDblClick(see TBaseVirtualTree.DoColumnDblClick Method , page 180)
Not documented.

DoColumnResize(see TBaseVirtualTree.DoColumnResize Method , page 180)
Not documented.

DoCompare(see TBaseVirtualTree.DoCompare Method , page 180)
Not documented.

DoCreateDataObject(see TBaseVirtualTree.DoCreateDataObject Method , page 180)
Not documented.

DoCreateDragManager(see TBaseVirtualTree.DoCreateDragManager Method , page 181)
Not documented.

DoCreateEditor(see TBaseVirtualTree.DoCreateEditor Method , page 181)
Not documented.

DoDragDrop(see TBaseVirtualTree.DoDragDrop Method , page 181)
Not documented.

DoDragExpand(see TBaseVirtualTree.DoDragExpand Method , page 181)

Not documented.

DoDragging(see TBaseVirtualTree.DoDragging Method , page 182)
Internal method which handles drag' drop.

DoDragOver(see TBaseVirtualTree.DoDragOver Method , page 182)
Not documented.

DoEdit(see TBaseVirtualTree.DoEdit Method , page 182)
Initiates editing of the currently set focused column and edit node.

DoEndDrag(see TBaseVirtualTree.DoEndDrag Method , page 182)
Not documented.

DoEndEdit(see TBaseVirtualTree.DoEndEdit Method , page 183)
Stops the current edit operation and takes over the new content.

DoExpanded(see TBaseVirtualTree.DoExpanded Method , page 183)
Not documented.

DoExpanding(see TBaseVirtualTree.DoExpanding Method , page 183)
Not documented.

DoFocusChange(see TBaseVirtualTree.DoFocusChange Method , page 184)
Not documented.

DoFocusChanging(see TBaseVirtualTree.DoFocusChanging Method , page 184)
Not documented.

DoFocusNode(see TBaseVirtualTree.DoFocusNode Method , page 184)
Internal method to set the focused node.

DoFreeNode(see TBaseVirtualTree.DoFreeNode Method , page 184)
Not documented.

DoGetAnimationType(see TBaseVirtualTree.DoGetAnimationType Method , page 185)
Determines the type of animation to be used.

DoGetCursor(see TBaseVirtualTree.DoGetCursor Method , page 185)
Not documented.

DoGetHeaderCursor(see TBaseVirtualTree.DoGetHeaderCursor Method , page 185)
Not documented.

DoGetImageIndex(see TBaseVirtualTree.DoGetImageIndex Method , page 186)
Not documented.

DoGetLineStyle(see TBaseVirtualTree.DoGetLineStyle Method , page 186)
Not documented.

DoGetNodeHint(see TBaseVirtualTree.DoGetNodeHint Method , page 186)
Not documented.

DoGetNodeTooltip(see TBaseVirtualTree.DoGetNodeTooltip Method , page 186)
Not documented.

DoGetNodeWidth(see TBaseVirtualTree.DoGetNodeWidth Method , page 187)
Overridable method which always retuns 0.

DoGetPopupMenu(see TBaseVirtualTree.DoGetPopupMenu Method , page 187)
Overridable method which triggers the OnGetPopup event.

DoGetUserClipboardFormats(see TBaseVirtualTree.DoGetUserClipboardFormats Method , page 187)
Not documented.

DoHeaderClick(see TBaseVirtualTree.DoHeaderClick Method , page 187)
Not documented.

DoHeaderDblClick(see TBaseVirtualTree.DoHeaderDblClick Method , page 188)
Not documented.

DoHeaderDragged(see TBaseVirtualTree.DoHeaderDragged Method , page 188)
Not documented.

DoHeaderDraggedOut(see TBaseVirtualTree.DoHeaderDraggedOut Method , page 188)
Not documented.

DoHeaderDragging(see TBaseVirtualTree.DoHeaderDragging Method , page 189)
Not documented.

DoHeaderDraw(see TBaseVirtualTree.DoHeaderDraw Method , page 189)
Not documented.

DoHeaderDrawQueryElements(see TBaseVirtualTree.DoHeaderDrawQueryElements Method , page 189)
Not documented.

DoHeaderMouseDown(see TBaseVirtualTree.DoHeaderMouseDown Method , page 189)
Not documented.

DoHeaderMouseMove(see TBaseVirtualTree.DoHeaderMouseMove Method , page 190)
Not documented.

DoHeaderMouseUp(see TBaseVirtualTree.DoHeaderMouseUp Method , page 190)
Not documented.

DoHotChange(see TBaseVirtualTree.DoHotChange Method , page 190)
Not documented.

DoIncrementalSearch(see TBaseVirtualTree.DoIncrementalSearch Method , page 190)
Not documented.

DoInitChildren(see TBaseVirtualTree.DoInitChildren Method , page 191)
Not documented.

DoInitNode(see TBaseVirtualTree.DoInitNode Method , page 191)
Not documented.

DoKeyAction(see TBaseVirtualTree.DoKeyAction Method , page 191)
Not documented.

DoLoadUserData(see TBaseVirtualTree.DoLoadUserData Method , page 191)
Not documented.

DoMeasureItem(see TBaseVirtualTree.DoMeasureItem Method , page 192)
Not documented.

DoNodeCopied(see TBaseVirtualTree.DoNodeCopied Method , page 192)
Not documented.

DoNodeCopying(see TBaseVirtualTree.DoNodeCopying Method , page 192)
Not documented.

DoNodeMoved(see TBaseVirtualTree.DoNodeMoved Method , page 192)
Not documented.

DoNodeMoving(see TBaseVirtualTree.DoNodeMoving Method , page 193)
Not documented.

DoPaintBackground(see TBaseVirtualTree.DoPaintBackground Method , page 193)
Not documented.

DoPaintDropMark(see TBaseVirtualTree.DoPaintDropMark Method , page 193)
Overridable method which draws the small line on top of a nodes image depending on the current drop state.

DoPaintNode(see TBaseVirtualTree.DoPaintNode Method , page 193)
Overridable method which does nothing.

DoPopupMenu(see TBaseVirtualTree.DoPopupMenu Method , page 194)
Overridable method which shows the popup menu for the given node.

DoRenderOLEData(see TBaseVirtualTree.DoRenderOLEData Method , page 194)
Not documented.

DoReset(see TBaseVirtualTree.DoReset Method , page 194)
Not documented.

DoSaveUserData(see TBaseVirtualTree.DoSaveUserData Method , page 194)
Not documented.

DoScroll(see TBaseVirtualTree.DoScroll Method , page 195)
Overridable method which triggers the OnScroll(see TBaseVirtualTree.OnScroll Event, page 152) event.

DoSetOffsetXY(see TBaseVirtualTree.DoSetOffsetXY Method , page 195)

Internal core routine to set the tree's scroll position.

DoShowScrollbar(see TBaseVirtualTree.DoShowScrollbar Method , page 195)
Not documented.

DoStartDrag(see TBaseVirtualTree.DoStartDrag Method , page 196)
Not documented.

DoStateChange(see TBaseVirtualTree.DoStateChange Method , page 196)
Not documented.

DoStructureChange(see TBaseVirtualTree.DoStructureChange Method , page 196)
Not documented.

DoTimerScroll(see TBaseVirtualTree.DoTimerScroll Method , page 196)
Callback method which is triggered whenever the scroll timer fires.

DoUpdating(see TBaseVirtualTree.DoUpdating Method , page 197)
Not documented.

DoValidateCache(see TBaseVirtualTree.DoValidateCache Method , page 197)
Not documented.

DragCanceled(see TBaseVirtualTree.DragCanceled Method , page 197)
Called by the VCL when a drag'n drop operation was canceled by the user.

DragDrop(see TBaseVirtualTree.DragDrop Method , page 197)
Helper method, which is used when a drag operation is finished.

DragEnter(see TBaseVirtualTree.DragEnter Method , page 198)
Not documented.

DragFinished(see TBaseVirtualTree.DragFinished Method , page 198)
Called when a drag operation is finished (accepted or cancelled).

Dragging(see TBaseVirtualTree.Dragging Method , page 198)
Returns true if a drag'n drop operation is in progress.

DragLeave(see TBaseVirtualTree.DragLeave Method , page 199)
Not documented.

DragOver(see TBaseVirtualTree.DragOver Method , page 199)
Not documented.

DrawDottedHLine(see TBaseVirtualTree.DrawDottedHLine Method , page 199)
Not documented.

DrawDottedVLine(see TBaseVirtualTree.DrawDottedVLine Method , page 199)
Not documented.

EditNode(see TBaseVirtualTree.EditNode Method , page 200)
Starts editing the given node if allowed to.

EndEditNode(see TBaseVirtualTree.EndEditNode Method , page 200)
Stops node editing if it was started before.

EndSynch(see TBaseVirtualTree.EndSynch Method , page 200)
Counterpart to BeginSynch(see TBaseVirtualTree.BeginSynch Method , page 164).

EndUpdate(see TBaseVirtualTree.EndUpdate Method , page 201)
Resets the update lock set by BeginUpdate(see TBaseVirtualTree.BeginUpdate Method , page 164).

ExecuteAction(see TBaseVirtualTree.ExecuteAction Method , page 201)
Not documented.

FindNodeInSelection(see TBaseVirtualTree.FindNodeInSelection Method , page 201)
Helper method to find the given node in the current selection.

FinishChunkHeader(see TBaseVirtualTree.FinishChunkHeader Method , page 201)
Not documented.

FinishCutOrCopy(see TBaseVirtualTree.FinishCutOrCopy Method , page 202)
Stops any pending cut or copy clipboard operation.

FlushClipboard(see TBaseVirtualTree.FlushClipboard Method , page 202)
Renders all pending clipboard data.

FontChanged(see TBaseVirtualTree.FontChanged Method , page 202)
Not documented.

FullCollapse(see TBaseVirtualTree.FullCollapse Method , page 203)
Collapses all nodes in the tree.

FullExpand(see TBaseVirtualTree.FullExpand Method , page 203)
Expands all nodes in the tree.

GetBorderDimensions(see TBaseVirtualTree.GetBorderDimensions Method , page 203)
Not documented.

GetCheckImage(see TBaseVirtualTree.GetCheckImage Method , page 203)
Not documented.

GetCheckImageListFor(see TBaseVirtualTree.GetCheckImageListFor Method , page 204)
Not documented.

GetColumnClass(see TBaseVirtualTree.GetColumnClass Method , page 204)
Returns the class to be used to manage columns in the tree.

GetControlsAlignment(see TBaseVirtualTree.GetControlsAlignment Method , page 204)
Not documented.

GetDisplayRect(see TBaseVirtualTree.GetDisplayRect Method , page 205)
Returns the visible region used by the given node in client coordinates.

GetFirst(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstChecked(see TBaseVirtualTree.GetFirstChecked Method , page 206)
Not documented.

GetFirstChild(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstCutCopy(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstInitialized(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstSelected(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisible(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleChild(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleChildNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetFirstVisibleNoInit(see TBaseVirtualTree.GetFirst Method , page 205)
Group of node navigation functions.

GetHeaderClass(see TBaseVirtualTree.GetHeaderClass Method , page 206)
Returns the header class to be used by the tree.

GetHintWindowClass(see TBaseVirtualTree.GetHintWindowClass Method , page 206)
Not documented.

GetHitTestInfoAt(see TBaseVirtualTree.GetHitTestInfoAt Method , page 206)
Returns information about the node at the given position.

GetImageIndex(see TBaseVirtualTree.GetImageIndex Method , page 207)
Not documented.

GetLast(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastChild(see TBaseVirtualTree.GetLast Method , page 207)

Group of node navigation functions.

GetLastChildNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastInitialized(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisible(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleChild(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleChildNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetLastVisibleNoInit(see TBaseVirtualTree.GetLast Method , page 207)
Group of node navigation functions.

GetMaxColumnWidth(see TBaseVirtualTree.GetMaxColumnWidth Method , page 208)
Returns the width of the largest node in the given column.

GetMaxRightExtend(see TBaseVirtualTree.GetMaxRightExtend Method , page 208)
Determines the maximum with of the currently visible part of the tree.

GetNativeClipboardFormats(see TBaseVirtualTree.GetNativeClipboardFormats Method , page 208)
Used to let descendants and the application add their own supported clipboard formats.

GetNext(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextChecked(see TBaseVirtualTree.GetNextChecked Method , page 209)
Not documented.

GetNextCutCopy(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextInitialized(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextSelected(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextSibling(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisible(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleSibling(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNextVisibleSiblingNoInit(see TBaseVirtualTree.GetNext Method , page 209)
Group of node navigation functions.

GetNodeAt(see TBaseVirtualTree.GetNodeAt Method (Integer, Integer), page 210)
Not documented.

GetNodeData(see TBaseVirtualTree.GetNodeData Method , page 210)
Returns the address of the user data area of the given node.

GetNodeLevel(see TBaseVirtualTree.GetNodeLevel Method , page 210)
Returns the indentation level of the given node.

GetOptionsClass(see TBaseVirtualTree.GetOptionsClass Method , page 211)
Customization helper to determine which options class the tree should use.

GetPrevious(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousInitialized(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousSibling(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisible(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleSibling(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetPreviousVisibleSiblingNoInit(see TBaseVirtualTree.GetPrevious Method , page 212)
Group of node navigation functions.

GetSortedCutCopySet(see TBaseVirtualTree.GetSortedCutCopySet Method , page 212)
Returns a sorted list of nodes, which are marked for s cut or copy clipboard operation.

GetSortedSelection(see TBaseVirtualTree.GetSortedSelection Method , page 213)
Returns a sorted list of all currently selected nodes.

GetTextInfo(see TBaseVirtualTree.GetTextInfo Method , page 213)
Helper method for node editors, hints etc.

GetTreeFromDataObject(see TBaseVirtualTree.GetTreeFromDataObject Method , page 213)
OLE drag'n drop and clipboard support method.

GetTreeRect(see TBaseVirtualTree.GetTreeRect Method , page 214)
Returns the size of the virtual tree image.

GetVisibleParent(see TBaseVirtualTree.GetVisibleParent Method , page 214)
Returns the first (nearest) parent node, which is visible.

HandleHotTrack(see TBaseVirtualTree.HandleHotTrack Method , page 214)
Not documented.

HandleIncrementalSearch(see TBaseVirtualTree.HandleIncrementalSearch Method , page 214)
Not documented.

HandleMouseDblClick(see TBaseVirtualTree.HandleMouseDblClick Method , page 215)
Not documented.

HandleMouseDown(see TBaseVirtualTree.HandleMouseDown Method , page 215)
Not documented.

HandleMouseUp(see TBaseVirtualTree.HandleMouseUp Method , page 215)
Not documented.

HasAsParent(see TBaseVirtualTree.HasAsParent Method , page 215)
Determines if the given node has got another node as one of its parents.

HasImage(see TBaseVirtualTree.HasImage Method , page 216)
Not documented.

HasPopupMenu(see TBaseVirtualTree.HasPopupMenu Method , page 216)
Determines whether there is a pop up menu assigned to the tree.

InitChildren(see TBaseVirtualTree.InitChildren Method , page 216)
Not documented.

InitNode(see TBaseVirtualTree.InitNode Method , page 217)
Not documented.

InsertNode(see TBaseVirtualTree.InsertNode Method , page 217)
Inserts a new node and returns it to the caller.

InternalAddFromStream(see TBaseVirtualTree.InternalAddFromStream Method , page 217)

Not documented.

InternalAddToSelection(see TBaseVirtualTree.InternalAddToSelection Method (PVirtualNode, Boolean), page 218)
Not documented.

InternalCacheNode(see TBaseVirtualTree.InternalCacheNode Method , page 218)
Not documented.

InternalClearSelection(see TBaseVirtualTree.InternalClearSelection Method , page 218)
Not documented.

InternalConnectNode(see TBaseVirtualTree.InternalConnectNode Method , page 219)
Not documented.

InternalData(see TBaseVirtualTree.InternalData Method , page 219)
Returns the address of the internal data for a tree class.

InternalDisconnectNode(see TBaseVirtualTree.InternalDisconnectNode Method , page 219)
Not documented.

InternalRemoveFromSelection(see TBaseVirtualTree.InternalRemoveFromSelection Method , page 220)
Not documented.

InvalidateCache(see TBaseVirtualTree.InvalidateCache Method , page 220)
Empties the internal node cache and marks it as invalid.

InvalidateChildren(see TBaseVirtualTree.InvalidateChildren Method , page 220)
Invalidates all children of the given node.

InvalidateColumn(see TBaseVirtualTree.InvalidateColumn Method , page 220)
Invalidates the client area part of a column.

InvalidateNode(see TBaseVirtualTree.InvalidateNode Method , page 221)
Invalidates the given node.

InvalidateToBottom(see TBaseVirtualTree.InvalidateToBottom Method , page 221)
Invalidates the client area starting with the top position of the given node.

InvertSelection(see TBaseVirtualTree.InvertSelection Method , page 221)
Inverts the current selection.

IsEditing(see TBaseVirtualTree.IsEditing Method , page 222)
Tells the caller whether the tree is currently in edit mode.

IsMouseSelecting(see TBaseVirtualTree.IsMouseSelecting Method , page 222)
Tell the caller whether the tree is currently in draw selection mode.

IterateSubtree(see TBaseVirtualTree.IterateSubtree Method , page 222)
Iterator method to go through all nodes of a given sub tree.

Loaded(see TBaseVirtualTree.Loaded Method , page 223)
Not documented.

LoadFromFile(see TBaseVirtualTree.LoadFromFile Method , page 223)
Loads previously streamed out tree data back in again.

LoadFromStream(see TBaseVirtualTree.LoadFromFile Method , page 223)
Loads previously streamed out tree data back in again.

MainColumnChanged(see TBaseVirtualTree.MainColumnChanged Method , page 223)
Not documented.

MarkCutCopyNodes(see TBaseVirtualTree.MarkCutCopyNodes Method , page 223)
Not documented.

MeasureItemHeight(see TBaseVirtualTree.MeasureItemHeight Method , page 224)
Not documented.

MouseMove(see TBaseVirtualTree.MouseMove Method , page 224)
Not documented.

MoveTo(see TBaseVirtualTree.MoveTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean), page 224)
Moves Source and all its child nodes to Target.

Notification(see TBaseVirtualTree.Notification Method , page 225)
Not documented.

OriginalWMNCPaint(see TBaseVirtualTree.OriginalWMNCPaint Method , page 225)
Not documented.

Paint(see TBaseVirtualTree.Paint Method , page 225)
TControl's Paint method used here to display the tree.

PaintCheckImage(see TBaseVirtualTree.PaintCheckImage Method , page 225)
Not documented.

PaintImage(see TBaseVirtualTree.PaintImage Method , page 226)
Not documented.

PaintNodeButton(see TBaseVirtualTree.PaintNodeButton Method , page 226)
Not documented.

PaintSelectionRectangle(see TBaseVirtualTree.PaintSelectionRectangle Method , page 226)
Not documented.

PaintTree(see TBaseVirtualTree.PaintTree Method , page 226)
Main paint routine for the tree image.

PaintTreeLines(see TBaseVirtualTree.PaintTreeLines Method , page 227)
Not documented.

PanningWindowProc(see TBaseVirtualTree.PanningWindowProc Method , page 227)
Not documented.

PasteFromClipboard(see TBaseVirtualTree.PasteFromClipboard Method , page 227)
Inserts the content of the clipboard into the tree.

PrepareDragImage(see TBaseVirtualTree.PrepareDragImage Method , page 228)
Not documented.

Print(see TBaseVirtualTree.Print Method , page 228)
Not documented.

ProcessDrop(see TBaseVirtualTree.ProcessDrop Method , page 228)
Helper method to ease OLE drag'n drop operations.

ProcessOLEData(see TBaseVirtualTree.ProcessOLEData Method , page 229)
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk(see TBaseVirtualTree.ReadChunk Method , page 229)
Not documented.

ReadNode(see TBaseVirtualTree.ReadNode Method , page 229)
Not documented.

RedirectFontChangeEvent(see TBaseVirtualTree.RedirectFontChangeEvent Method , page 230)
Not documented.

ReinitChildren(see TBaseVirtualTree.ReinitChildren Method , page 230)
Forces all child nodes of Node to be reinitialized.

ReinitNode(see TBaseVirtualTree.ReinitNode Method , page 230)
Forces a reinitialization of the given node.

RemoveFromSelection(see TBaseVirtualTree.RemoveFromSelection Method , page 230)
Removes the given node from the current selection.

RenderOLEData(see TBaseVirtualTree.RenderOLEData Method , page 231)
Renders pending OLE data.

RepaintNode(see TBaseVirtualTree.RepaintNode Method , page 231)
Causes the treeview to repaint the given node.

ResetNode(see TBaseVirtualTree.ResetNode Method , page 231)
Resets the given node to uninitialized.

ResetRangeAnchor(see TBaseVirtualTree.ResetRangeAnchor Method , page 232)
Not documented.

RestoreFontChangeEvent(see TBaseVirtualTree.RestoreFontChangeEvent Method , page 232)
Not documented.

SaveToFile(see TBaseVirtualTree.SaveToFile Method , page 232)

Saves the entire content of the tree into a file or stream.

SaveToStream(see TBaseVirtualTree.SaveToFile Method , page 232)
Saves the entire content of the tree into a file or stream.

ScrollIntoView(see TBaseVirtualTree.ScrollIntoView Method , page 232)
Scrolls the tree so that the given node comes in the client area.

SelectAll(see TBaseVirtualTree.SelectAll Method , page 233)
Selects all nodes in the tree.

SelectNodes(see TBaseVirtualTree.SelectNodes Method , page 233)
Selects a range of nodes.

SetBiDiMode(see TBaseVirtualTree.SetBiDiMode Method , page 233)
Not documented.

SetFocusedNodeAndColumn(see TBaseVirtualTree.SetFocusedNodeAndColumn Method , page 234)
Not documented.

SkipNode(see TBaseVirtualTree.SkipNode Method , page 234)
Not documented.

Sort(see TBaseVirtualTree.Sort Method , page 234)
Sorts the given node.

SortTree(see TBaseVirtualTree.SortTree Method , page 234)
Sorts the entire tree view.

StartWheelPanning(see TBaseVirtualTree.StartWheelPanning Method , page 235)
Not documented.

StopWheelPanning(see TBaseVirtualTree.StopWheelPanning Method , page 235)
Not documented.

StructureChange(see TBaseVirtualTree.StructureChange Method , page 235)
Not documented.

SuggestDropEffect(see TBaseVirtualTree.SuggestDropEffect Method , page 236)
Not documented.

ToggleNode(see TBaseVirtualTree.ToggleNode Method , page 236)
Changes a node's expand state to the opposite state.

ToggleSelection(see TBaseVirtualTree.ToggleSelection Method , page 236)
Toggles the selection state of a range of nodes.

UnselectNodes(see TBaseVirtualTree.UnselectNodes Method , page 236)
Deselects a range of nodes.

UpdateAction(see TBaseVirtualTree.UpdateAction Method , page 237)
Not documented.

UpdateDesigner(see TBaseVirtualTree.UpdateDesigner Method , page 237)
Not documented.

UpdateEditBounds(see TBaseVirtualTree.UpdateEditBounds Method , page 237)
Not documented.

UpdateHeaderRect(see TBaseVirtualTree.UpdateHeaderRect Method , page 237)
Not documented.

UpdateHorizontalScrollBar(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateScrollBars(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateVerticalScrollBar(see TBaseVirtualTree.UpdateScrollBars Method , page 238)
Applies changes to the horizontal and vertical scrollbars.

UpdateWindowAndDragImage(see TBaseVirtualTree.UpdateWindowAndDragImage Method , page 238)
Not documented.

UseRightToLeftReading(see TBaseVirtualTree.UseRightToLeftReading Method , page 238)
Helper method for right-to-left layout.

ValidateCache(see TBaseVirtualTree.ValidateCache Method , page 239)
Initiates the validation of the internal node cache.

ValidateChildren(see TBaseVirtualTree.ValidateChildren Method , page 239)
Validates all children of a given node.

ValidateNode(see TBaseVirtualTree.ValidateNode Method , page 239)
Validates a given node.

ValidateNodeDataSize(see TBaseVirtualTree.ValidateNodeDataSize Method , page 239)
Helper method for node data size initalization.

WndProc(see TBaseVirtualTree.WndProc Method , page 240)
Redirected window procedure to do some special processing.

WriteChunks(see TBaseVirtualTree.WriteChunks Method , page 240)
Writes the core chunks for the given node to the given stream.

WriteNode(see TBaseVirtualTree.WriteNode Method , page 241)
Writes the cover (envelop) chunk for the given node to the given stream.

Properties

Action(see TVirtualStringTree.Action Property, page 430)
Not documented.

Align(see TVirtualStringTree.Align Property, page 430)
Not documented.

Alignment(see TVirtualStringTree.Alignment Property, page 430)
Determines the horizontal alignment of text if no columns are defined.

Anchors(see TVirtualStringTree.Anchors Property, page 430)
Not documented.

AnimationDuration(see TVirtualStringTree.AnimationDuration Property, page 431)
Determines the maximum duration the tree can use to play an animation.

AutoExpandDelay(see TVirtualStringTree.AutoExpandDelay Property, page 431)
Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay(see TVirtualStringTree.AutoScrollDelay Property, page 431)
Time which determines when auto scrolling should start.

AutoScrollInterval(see TVirtualStringTree.AutoScrollInterval Property, page 432)
Time interval between scroll events when doing auto scroll.

Background(see TVirtualStringTree.Background Property, page 432)
Holds a background image for the tree.

BackgroundOffsetX(see TVirtualStringTree.BackgroundOffsetX Property, page 432)
Horizontal offset of the background image.

BackgroundOffsetY(see TVirtualStringTree.BackgroundOffsetY Property, page 432)
Vertical offset of the background image.

BevelEdges(see TVirtualStringTree.BevelEdges Property, page 433)
Not documented.

BevelInner(see TVirtualStringTree.BevelInner Property, page 433)
Not documented.

BevelKind(see TVirtualStringTree.BevelKind Property, page 433)
Not documented.

BevelOuter(see TVirtualStringTree.BevelOuter Property, page 433)
Not documented.

BevelWidth(see TVirtualStringTree.BevelWidth Property, page 434)
Not documented.

BiDiMode(see TVirtualStringTree.BiDiMode Property, page 434)
Not documented.

BorderStyle(see TVirtualStringTree.BorderStyle Property, page 434)
Same as TForm.BorderStyle.

BorderWidth(see TVirtualStringTree.BorderWidth Property, page 434)
Not documented.

ButtonFillMode(see TVirtualStringTree.ButtonFillMode Property, page 435)
Determines how to fill the background of the node buttons.

ButtonStyle(see TVirtualStringTree.ButtonStyle Property, page 435)
Determines the look of node buttons.

Canvas(see TVirtualStringTree.Canvas Property, page 435)
Not documented.

ChangeDelay(see TVirtualStringTree.ChangeDelay Property, page 436)
Time which determines when the OnChange event should be triggered after the actual change event.

CheckImageKind(see TVirtualStringTree.CheckImageKind Property, page 436)
Determines which images should be used for checkboxes and radio buttons.

ClipboardFormats(see TVirtualStringTree.ClipboardFormats Property, page 436)
Special class to keep a list of clipboard format descriptions.

Color(see TVirtualStringTree.Color Property, page 437)
Not documented.

Colors(see TVirtualStringTree.Colors Property, page 437)
A collection of colors used in the tree.

Constraints(see TVirtualStringTree.Constraints Property, page 437)
Not documented.

Ctl3D(see TVirtualStringTree.Ctl3D Property, page 437)
Not documented.

CustomCheckImages(see TVirtualStringTree.CustomCheckImages Property, page 438)
Assign your own image list to get the check images you like most.

DefaultNodeHeight(see TVirtualStringTree.DefaultNodeHeight Property, page 438)
Read or set the height new nodes get as initial value.

DefaultPasteMode(see TVirtualStringTree.DefaultPasteMode Property, page 438)
Read or set the value, which determines where to add pasted nodes to.

DefaultText(see TVirtualStringTree.DefaultText Property, page 439)
Not documented.

DragCursor(see TVirtualStringTree.DragCursor Property, page 439)
Not documented.

DragHeight(see TVirtualStringTree.DragHeight Property, page 439)
Read or set the vertical limit of the internal drag image.

DragImageKind(see TVirtualStringTree.DragImageKind Property, page 439)
Read or set what should be shown in the drag image.

DragKind(see TVirtualStringTree.DragKind Property, page 440)
Not documented.

DragMode(see TVirtualStringTree.DragMode Property, page 440)
Not documented.

DragOperations(see TVirtualStringTree.DragOperations Property, page 440)
Read or set which drag operations may be allowed in the tree.

DragType(see TVirtualStringTree.DragType Property, page 440)
Read or set which subsystem should be used for dragging.

DragWidth(see TVirtualStringTree.DragWidth Property, page 441)
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode(see TVirtualStringTree.DrawSelectionMode Property, page 441)
Read or set how multiselection with the mouse is to be visualized.

EditDelay(see TVirtualStringTree.EditDelay Property, page 441)
Read or set the maximum time between two single clicks on the same node, which should start node editing.

Enabled(see TVirtualStringTree.Enabled Property, page 442)

Not documented.

Font(see TVirtualStringTree.Font Property, page 442)
Same as TWinControl.Font.

Header(see TVirtualStringTree.Header Property, page 442)
Provides access to the header instance.

HintAnimation(see TVirtualStringTree.HintAnimation Property, page 443)
Read or set the current hint animation type.

HintMode(see TVirtualStringTree.HintMode Property, page 443)
Read or set what type of hint you want for the tree view.

HotCursor(see TVirtualStringTree.HotCursor Property, page 443)
Read or set which cursor should be used for hot nodes.

Images(see TVirtualStringTree.Images Property, page 444)
Read or set the tree's normal image list.

IncrementalSearch(see TVirtualStringTree.IncrementalSearch Property, page 444)
Read or set the current incremental search mode.

IncrementalSearchDirection(see TVirtualStringTree.IncrementalSearchDirection Property, page 444)
Read or set the direction to be used for incremental search.

IncrementalSearchStart(see TVirtualStringTree.IncrementalSearchStart Property, page 445)
Read or set where to start incremental search.

IncrementalSearchTimeout(see TVirtualStringTree.IncrementalSearchTimeout Property, page 445)
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Indent(see TVirtualStringTree.Indent Property, page 445)
Read or set the indentation amount for node levels.

LineMode(see TVirtualStringTree.LineMode Property, page 446)
Read or set the mode of the tree lines.

LineStyle(see TVirtualStringTree.LineStyle Property, page 446)
Read or set the mode of the tree lines.

Margin(see TVirtualStringTree.Margin Property, page 446)
Read or set the tree's node margin.

NodeAlignment(see TVirtualStringTree.NodeAlignment Property, page 446)
Read or set the node alignment value.

NodeDataSize(see TVirtualStringTree.NodeDataSize Property, page 447)
Read or set the extra data size for each node.

OnClick(see TVirtualStringTree.OnClick Property, page 451)
Not documented.

OnDblClick(see TVirtualStringTree.OnDblClick Property, page 454)
Not documented.

OnEndDock(see TVirtualStringTree.OnEndDock Property, page 457)
Not documented.

OnEndDrag(see TVirtualStringTree.OnEndDrag Property, page 458)
Not documented.

OnEnter(see TVirtualStringTree.OnEnter Property, page 458)
Not documented.

OnExit(see TVirtualStringTree.OnExit Property, page 458)
Not documented.

OnKeyDown(see TVirtualStringTree.OnKeyDown Property, page 470)
Not documented.

OnKeyPress(see TVirtualStringTree.OnKeyPress Property, page 470)
Not documented.

OnKeyUp(see TVirtualStringTree.OnKeyUp Property, page 470)
Not documented.

OnMouseDown(see TVirtualStringTree.OnMouseDown Property, page 471)
Not documented.

OnMouseMove(see TVirtualStringTree.OnMouseMove Property, page 471)
Not documented.

OnMouseUp(see TVirtualStringTree.OnMouseUp Property, page 472)
Not documented.

OnMouseWheel(see TVirtualStringTree.OnMouseWheel Property, page 472)
Not documented.

OnResize(see TVirtualStringTree.OnResize Property, page 475)
Not documented.

OnStartDock(see TVirtualStringTree.OnStartDock Property, page 477)
Not documented.

OnStartDrag(see TVirtualStringTree.OnStartDrag Property, page 477)
Not documented.

ParentBiDiMode(see TVirtualStringTree.ParentBiDiMode Property, page 478)
Not documented.

ParentColor(see TVirtualStringTree.ParentColor Property, page 479)
Not documented.

ParentCtl3D(see TVirtualStringTree.ParentCtl3D Property, page 479)
Not documented.

ParentFont(see TVirtualStringTree.ParentFont Property, page 479)
Not documented.

ParentShowHint(see TVirtualStringTree.ParentShowHint Property, page 479)
Not documented.

PopupMenu(see TVirtualStringTree.PopupMenu Property, page 480)
Not documented.

RootNodeCount(see TVirtualStringTree.RootNodeCount Property, page 480)
Read or set the number of nodes on the top level.

ScrollBarOptions(see TVirtualStringTree.ScrollBarOptions Property, page 480)
Reference to the scroll bar options class.

SelectionBlendFactor(see TVirtualStringTree.SelectionBlendFactor Property, page 480)
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

SelectionCurveRadius(see TVirtualStringTree.SelectionCurveRadius Property, page 481)
Read or set the current corner radius for node selection rectangles.

ShowHint(see TVirtualStringTree.ShowHint Property, page 481)
Not documented.

StateImages(see TVirtualStringTree.StateImages Property, page 481)
Reference to the images list which is used for the state images.

TabOrder(see TVirtualStringTree.TabOrder Property, page 482)
Not documented.

TabStop(see TVirtualStringTree.TabStop Property, page 482)
Not documented.

TextMargin(see TVirtualStringTree.TextMargin Property, page 482)
Read or set the distance of the node caption to its borders.

TreeOptions(see TVirtualStringTree.TreeOptions Property, page 483)
Reference to the tree's options.

Visible(see TVirtualStringTree.Visible Property, page 483)
Not documented.

WantTabs(see TVirtualStringTree.WantTabs Property, page 483)
Read or set whether the tree wants to process tabs on its own.

TCustomVirtualStringTree Class

DefaultText(see TCustomVirtualStringTree.DefaultText Property, page 295)
Not documented.

EllipsisWidth(see TCustomVirtualStringTree.EllipsisWidth Property, page 295)
Not documented.

Text(see TCustomVirtualStringTree.Text Property, page 298)
Not documented.

TreeOptions(see TCustomVirtualStringTree.TreeOptions Property, page 298)
Reference to the tree's options.

TBaseVirtualTree Class

Alignment(see TBaseVirtualTree.Alignment Property, page 107)
Determines the horizontal alignment of text if no columns are defined.

AnimationDuration(see TBaseVirtualTree.AnimationDuration Property, page 107)
Determines the maximum duration the tree can use to play an animation.

AutoExpandDelay(see TBaseVirtualTree.AutoExpandDelay Property, page 107)
Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay(see TBaseVirtualTree.AutoScrollDelay Property, page 108)
Time which determines when auto scrolling should start.

AutoScrollInterval(see TBaseVirtualTree.AutoScrollInterval Property, page 108)
Time interval between scroll events when doing auto scroll.

Background(see TBaseVirtualTree.Background Property, page 108)
Holds a background image for the tree.

BackgroundOffsetX(see TBaseVirtualTree.BackgroundOffsetX Property, page 109)
Horizontal offset of the background image.

BackgroundOffsetY(see TBaseVirtualTree.BackgroundOffsetY Property, page 109)
Vertical offset of the background image.

BorderStyle(see TBaseVirtualTree.BorderStyle Property, page 109)
Same as TForm.BorderStyle.

ButtonFillMode(see TBaseVirtualTree.ButtonFillMode Property, page 109)
Determines how to fill the background of the node buttons.

ButtonStyle(see TBaseVirtualTree.ButtonStyle Property, page 110)
Determines the look of node buttons.

ChangeDelay(see TBaseVirtualTree.ChangeDelay Property, page 110)
Time which determines when the OnChange(see TBaseVirtualTree.OnChange Event, page 131) event should be triggered after the actual change event.

CheckImageKind(see TBaseVirtualTree.CheckImageKind Property, page 110)
Determines which images should be used for checkboxes and radio buttons.

CheckImages(see TBaseVirtualTree.CheckImages Property, page 111)
Not documented.

CheckState(see TBaseVirtualTree.CheckState Property, page 111)
Read or set the check state of a node.

CheckType(see TBaseVirtualTree.CheckType Property, page 111)
Read or set the check type of a node.

ChildCount(see TBaseVirtualTree.ChildCount Property, page 111)
Read or set the number of child nodes of a node.

ChildrenInitialized(see TBaseVirtualTree.ChildrenInitialized Property, page 112)
Read whether a node's child count has been initialized already.

ClipboardFormats(see TBaseVirtualTree.ClipboardFormats Property, page 112)
Special class to keep a list of clipboard format descriptions.

Colors(see TBaseVirtualTree.Colors Property, page 112)
A collection of colors used in the tree.

CustomCheckImages(see TBaseVirtualTree.CustomCheckImages Property, page 113)
Assign your own image list to get the check images you like most.

DefaultNodeHeight(see TBaseVirtualTree.DefaultNodeHeight Property, page 113)
Read or set the height new nodes get as initial value.

DefaultPasteMode(see TBaseVirtualTree.DefaultPasteMode Property, page 113)
Read or set the value, which determines where to add pasted nodes to.

DragHeight(see TBaseVirtualTree.DragHeight Property, page 114)
Read or set the vertical limit of the internal drag image.

DragImage(see TBaseVirtualTree.DragImage Property, page 114)
Holds the instance of the internal drag image.

DragImageKind(see TBaseVirtualTree.DragImageKind Property, page 114)
Read or set what should be shown in the drag image.

DragManager(see TBaseVirtualTree.DragManager Property, page 114)
Holds the reference to the internal drag manager.

DragOperations(see TBaseVirtualTree.DragOperations Property, page 115)
Read or set which drag operations may be allowed in the tree.

DragSelection(see TBaseVirtualTree.DragSelection Property, page 115)
Keeps a temporary list of nodes during drag'n drop.

DragType(see TBaseVirtualTree.DragType Property, page 115)
Read or set which subsystem should be used for dragging(see TBaseVirtualTree.Dragging Method , page 198).

DragWidth(see TBaseVirtualTree.DragWidth Property, page 116)
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode(see TBaseVirtualTree.DrawSelectionMode Property, page 116)
Read or set how multiselection with the mouse is to be visualized.

DropTargetNode(see TBaseVirtualTree.DropTargetNode Property, page 116)
Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

EditColumn(see TBaseVirtualTree.EditColumn Property, page 117)
Not documented.

EditDelay(see TBaseVirtualTree.EditDelay Property, page 117)
Read or set the maximum time between two single clicks on the same node, which should start node editing.

EditLink(see TBaseVirtualTree.EditLink Property, page 117)
Keeps a reference to the internal edit link during a node edit operation.

Expanded(see TBaseVirtualTree.Expanded Property, page 118)
Read or set the expanded state of a particular node.

FocusedColumn(see TBaseVirtualTree.FocusedColumn Property, page 118)
Read or set the currently focused collumn.

FocusedNode(see TBaseVirtualTree.FocusedNode Property, page 118)
Read or set the currently focused node.

Font(see TBaseVirtualTree.Font Property, page 119)
Same as TWinControl.Font.

FullyVisible(see TBaseVirtualTree.FullyVisible Property, page 119)
Read or set whether a node is fully visible or not.

HasChildren(see TBaseVirtualTree.HasChildren Property, page 119)
Read or set whether a node has got children.

Header(see TBaseVirtualTree.Header Property, page 120)
Provides access to the header instance.

HeaderRect(see TBaseVirtualTree.HeaderRect Property, page 120)
Returns the non-client-area rectangle used for the header.

HintAnimation(see TBaseVirtualTree.HintAnimation Property, page 120)
Read or set the current hint animation type.

HintMode(see TBaseVirtualTree.HintMode Property, page 121)

Read or set what type of hint you want for the tree view.

HotCursor(see TBaseVirtualTree.HotCursor Property, page 121)
Read or set which cursor should be used for hot nodes.

HotNode(see TBaseVirtualTree.HotNode Property, page 121)
Read, which node is currently the hot node.

Images(see TBaseVirtualTree.Images Property, page 122)
Read or set the tree's normal image list.

IncrementalSearch(see TBaseVirtualTree.IncrementalSearch Property, page 122)
Read or set the current incremental search mode.

IncrementalSearchDirection(see TBaseVirtualTree.IncrementalSearchDirection Property, page 122)
Read or set the direction to be used for incremental search.

IncrementalSearchStart(see TBaseVirtualTree.IncrementalSearchStart Property, page 123)
Read or set where to start incremental search.

IncrementalSearchTimeout(see TBaseVirtualTree.IncrementalSearchTimeout Property, page 123)
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Indent(see TBaseVirtualTree.Indent Property, page 123)
Read or set the indentation amount for node levels.

IsDisabled(see TBaseVirtualTree.IsDisabled Property, page 124)
Read or set the enabled state of the given node.

IsVisible(see TBaseVirtualTree.IsVisible Property, page 124)
Read or set the visibility state of the given node.

LastClickPos(see TBaseVirtualTree.LastClickPos Property, page 124)
Used for retained drag start and wheel mouse scrolling.

LastDropMode(see TBaseVirtualTree.LastDropMode Property, page 125)
Read how the last drop operation finished.

LineMode(see TBaseVirtualTree.LineMode Property, page 125)
Read or set the mode of the tree lines.

LineStyle(see TBaseVirtualTree.LineStyle Property, page 125)
Read or set the mode of the tree lines.

Margin(see TBaseVirtualTree.Margin Property, page 125)
Read or set the tree's node margin.

MultiLine(see TBaseVirtualTree.MultiLine Property, page 126)
Read or toggle the multiline feature for a given node.

NodeAlignment(see TBaseVirtualTree.NodeAlignment Property, page 126)
Read or set the node alignment value.

NodeDataSize(see TBaseVirtualTree.NodeDataSize Property, page 127)
Read or set the extra data size for each node.

NodeHeight(see TBaseVirtualTree.NodeHeight Property, page 127)
Read or set a node's height.

NodeParent(see TBaseVirtualTree.NodeParent Property, page 127)
Read or set a node's parent node.

OffsetX(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

OffsetXY(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

OffsetY(see TBaseVirtualTree.OffsetXY Property, page 128)
Read or set the tree's current horizontal and vertical scroll offsets.

RootNode(see TBaseVirtualTree.RootNode Property, page 153)
Reference to the internal root node which is the anchor of the entire tree node hierarchy.

RootNodeCount(see TBaseVirtualTree.RootNodeCount Property, page 153)
Read or set the number of nodes on the top level.

ScrollBarOptions(see TBaseVirtualTree.ScrollBarOptions Property, page 154)
Reference to the scroll bar options class.

SearchBuffer(see TBaseVirtualTree.SearchBuffer Property, page 154)
Current input string for incremental search.

Selected(see TBaseVirtualTree.Selected Property, page 154)
Property to modify or determine the selection state of a node.

SelectedCount(see TBaseVirtualTree.SelectedCount Property, page 155)
Contains the number of selected nodes.

SelectionBlendFactor(see TBaseVirtualTree.SelectionBlendFactor Property, page 155)
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

SelectionCurveRadius(see TBaseVirtualTree.SelectionCurveRadius Property, page 155)
Read or set the current corner radius for node selection rectangles.

StateImages(see TBaseVirtualTree.StateImages Property, page 156)
Reference to the images list which is used for the state images.

TextMargin(see TBaseVirtualTree.TextMargin Property, page 156)
Read or set the distance of the node caption to its borders.

TopNode(see TBaseVirtualTree.TopNode Property, page 157)
The top node is the node which is currently at the top border of the client area.

TotalCount(see TBaseVirtualTree.TotalCount Property, page 157)
Returns the number of nodes in the tree.

TotalInternalDataSize(see TBaseVirtualTree.TotalInternalDataSize Property, page 157)
Keeps the currently accumulated data size for one node.

TreeOptions(see TBaseVirtualTree.TreeOptions Property, page 158)
Reference to the tree's options.

TreeStates(see TBaseVirtualTree.TreeStates Property, page 158)
Property which keeps a set of flags which indicate current operation and states of the tree.

UpdateCount(see TBaseVirtualTree.UpdateCount Property, page 158)
Not documented.

VerticalAlignment(see TBaseVirtualTree.VerticalAlignment Property, page 158)
Used to set a node's vertical button aligment with regard to the entire node rectangle.

VisibleCount(see TBaseVirtualTree.VisibleCount Property, page 159)
Number of currently visible nodes.

VisiblePath(see TBaseVirtualTree.VisiblePath Property, page 159)
Property to set or determine a node parent's expand states.

WantTabs(see TBaseVirtualTree.WantTabs Property, page 159)
Read or set whether the tree wants to process tabs on its own.

Legend

published

Property

public

protected

read only

Event

Method

virtual

Class Hierarchy

File

VirtualTrees

10.1.16.1 TVirtualStringTree.Action Property

Not documented.

Pascal

```
property Action;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.2 TVirtualStringTree.Align Property

Not documented.

Pascal

```
property Align;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.3 TVirtualStringTree.Alignment Property

Determines the horizontal alignment of text if no columns are defined.

Pascal

```
property Alignment: TAlignment;
```

Description

This property is only used if there are no columns defined and applies only to the node captions. Right alignment means here the right client area border and left aligned means the node buttons/lines etc. (both less the text margin).

Class

TVirtualStringTree Class(see page 402)

10.1.16.4 TVirtualStringTree.Anchors Property

Not documented.

Pascal

```
property Anchors;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.5 TVirtualStringTree.AnimationDuration Property

Determines the maximum duration the tree can use to play an animation.

Pascal

```
property AnimationDuration: Cardinal;
```

Description

The value is specified in milliseconds and per default there are 200 ms as time frame, which is the recommended duration for such operations. On older systems (particularly Windows 95 and Windows 98) the animation process might not get enough CPU time to avoid expensive animations to finish properly. Still the animation loop tries to stay as close as possible to the given time.

Class

TVirtualStringTree Class(see page 402)

10.1.16.6 TVirtualStringTree.AutoExpandDelay Property

Time delay after which a node gets expanded if it is the current drop target.

Pascal

```
property AutoExpandDelay: Cardinal;
```

Description

This value is specified in milliseconds and determines when to expand a node if it is the current drop target. This value is only used if voAutoDropExpand in Options is set.

Class

TVirtualStringTree Class(see page 402)

10.1.16.7 TVirtualStringTree.AutoScrollDelay Property

Time which determines when auto scrolling should start.

Pascal

```
property AutoScrollDelay: Cardinal;
```

Description

Once the mouse pointer has been moved near to a border a timer is started using the interval specified by AutoScrollDelay. When the timer has fired auto scrolling starts provided it is enabled (see also TreeOptions). The value is specified in milliseconds.

Class

TVirtualStringTree Class(see page 402)

10.1.16.8 TVirtualStringTree.AutoScrollInterval Property

Time interval between scroll events when doing auto scroll.

Pascal

```
property AutoScrollInterval: TAutoScrollInterval;
```

Description

This property determines the speed how the tree is scrolled vertically or horizontally when auto scrolling is in progress. The value is given in milliseconds.

Class

TVirtualStringTree Class(see page 402)

10.1.16.9 TVirtualStringTree.Background Property

Holds a background image for the tree.

Pascal

```
property Background: TPicture;
```

Description

Virtual Treeview supports a fixed background image which does not scroll but can be adjusted by BackgroundOffsetX and BackgroundOffsetY.

Class

TVirtualStringTree Class(see page 402)

10.1.16.10 TVirtualStringTree.BackgroundOffsetX Property

Horizontal offset of the background image.

Pascal

```
property BackgroundOffsetX: Integer;
```

Description

Determines the horizontal offset of the left border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

TVirtualStringTree Class(see page 402)

10.1.16.11 TVirtualStringTree.BackgroundOffsetY Property

Vertical offset of the background image.

Pascal

```
property BackgroundOffsetY: Integer;
```

Description

Determines the vertical offset of the top border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

TVirtualStringTree Class(see page 402)

10.1.16.12 TVirtualStringTree.BevelEdges Property

Not documented.

Pascal

```
property BevelEdges;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.13 TVirtualStringTree.BevelInner Property

Not documented.

Pascal

```
property BevelInner;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.14 TVirtualStringTree.BevelKind Property

Not documented.

Pascal

```
property BevelKind;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.15 TVirtualStringTree.BevelOuter Property

Not documented.

Pascal

```
property BevelOuter;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.16 TVirtualStringTree.BevelWidth Property

Not documented.

Pascal

```
property BevelWidth;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.17 TVirtualStringTree.BiDiMode Property

Not documented.

Pascal

```
property BiDiMode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.18 TVirtualStringTree.BorderStyle Property

Same as TForm.BorderStyle.

Pascal

```
property BorderStyle: TBorderStyle;
```

Description

See TForm.BorderStyle.

Class

TVirtualStringTree Class(see page 402)

10.1.16.19 TVirtualStringTree.BorderWidth Property

Not documented.

Pascal

```
property BorderWidth;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.20 TVirtualStringTree.ButtonFillMode Property

Determines how to fill the background of the node buttons.

Pascal

```
property ButtonFillMode: TVTButtonFillMode;
```

Description

This property is used to specify how the interior of the little plus and minus node buttons should be drawn, if ButtonStyle is bsTriangle.

Class

TVirtualStringTree Class(see page 402)

10.1.16.21 TVirtualStringTree.ButtonStyle Property

Determines the look of node buttons.

Pascal

```
property ButtonStyle: TVTButtonStyle;
```

Description

Determines the look of node buttons.

Class

TVirtualStringTree Class(see page 402)

10.1.16.22 TVirtualStringTree.Canvas Property

Not documented.

Pascal

```
property Canvas;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.23 TVirtualStringTree.ChangeDelay Property

Time which determines when the OnChange event should be triggered after the actual change event.

Pascal

```
property ChangeDelay: Cardinal;
```

Description

In order to accumulate many quick changes in the tree you can use this delay value to specify after which wait time the OnChange event should occur. A value of 0 means to trigger OnChange immediately after the change (usually a selection or focus change) happend. Any value > 0 will start a timer which then triggers OnChange.

Note that there is the synchronous mode (started by BeginSynch) which effectively circumvents the change delay for the duration of the synchronous mode (stopped by EndSynch) regardless of the ChangeDelay setting.

Class

TVirtualStringTree Class(see page 402)

10.1.16.24 TVirtualStringTree.CheckImageKind Property

Determines which images should be used for checkboxes and radio buttons.

Pascal

```
property CheckImageKind: TCheckImageKind;
```

Description

CheckImageKind can be used to switch the image set, which should be used for the tree. Read the description about TCheckImageKind for a list of all images, which can be used. CheckImageKind can also be set to ckCustom, which allows to supply a customized set of images to the tree. In order to have that working you must assign an image list (TCustomImageList) to the CustomCheckImages property.

Class

TVirtualStringTree Class(see page 402)

10.1.16.25 TVirtualStringTree.ClipboardFormats Property

Special class to keep a list of clipboard format descriptions.

Pascal

```
property ClipboardFormats: TClipboardFormats;
```

Description

This TStringList descendant is used to keep a number of clipboard format descriptions, which are usually used to register clipboard formats with the system. Using a string list for this task allows to store enabled clipboard formats in the DFM.

Class

TVirtualStringTree Class(see page 402)

10.1.16.26 TVirtualStringTree.Color Property

Not documented.

Pascal

```
property Color;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.27 TVirtualStringTree.Colors Property

A collection of colors used in the tree.

Pascal

```
property Colors: TVTColors;
```

Description

This property holds an instance of the TVTColors class, which is used to customize many of the colors used in a tree. Placing them all in a specialized class helps organizing the colors in the object inspector and improves general management.

Class

TVirtualStringTree Class(see page 402)

10.1.16.28 TVirtualStringTree.Constraints Property

Not documented.

Pascal

```
property Constraints;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.29 TVirtualStringTree.Ctl3D Property

Not documented.

Pascal

```
property Ctl3D;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.30 TVirtualStringTree.CustomCheckImages Property

Assign your own image list to get the check images you like most.

Pascal

```
property CustomCheckImages: TCustomImageList;
```

Description

The CustomCheckImages property is used when custom check images are enabled (see also ckCustom in TCheckImageKind).

See Also

TCheckImageKind

Class

TVirtualStringTree Class(see page 402)

10.1.16.31 TVirtualStringTree.DefaultNodeHeight Property

Read or set the height new nodes get as initial value.

Pascal

```
property DefaultNodeHeight: Cardinal;
```

Description

This property allows to read the current initial height for new nodes and to set a new value. Note that changing the property value does not change the height of existing nodes. Only new nodes are affected.

Class

TVirtualStringTree Class(see page 402)

10.1.16.32 TVirtualStringTree.DefaultPasteMode Property

Read or set the value, which determines where to add pasted nodes to.

Pascal

```
property DefaultPasteMode: TVTNodeAttachMode;
```

Description

The default paste mode is an attach mode, which is used when pasting data from the clipboard into the tree. Usually, you will want new nodes to be added as child nodes to the currently focused node (and this is also the default value), but you can also specify to add nodes only as siblings.

See Also

TVTNodeAttachMode

Class

TVirtualStringTree Class(see page 402)

10.1.16.33 TVirtualStringTree.DefaultText Property

Not documented.

Pascal

```
property DefaultText: WideString;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.34 TVirtualStringTree.DragCursor Property

Not documented.

Pascal

```
property DragCursor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.35 TVirtualStringTree.DragHeight Property

Read or set the vertical limit of the internal drag image.

Pascal

```
property DragHeight: Integer;
```

Description

The DragHeight property (as well as the DragWidth property) are only for compatibility reason in the tree. If a platform does not support the IDropTargetHelper interface (Windows 9x/Me, Windows NT 4.0) then Virtual Treeview uses its own implementation of a DragImage. Since displaying a translucent drag image is performance hungry you should limit the image size shown for the drag operation.

Class

TVirtualStringTree Class(see page 402)

10.1.16.36 TVirtualStringTree.DragImageKind Property

Read or set what should be shown in the drag image.

Pascal

```
property DragImageKind: TVTDragImageKind;
```

Description

DragImageKind allows to switch parts of the drag image off and on.

Class

TVirtualStringTree Class(see page 402)

10.1.16.37 TVirtualStringTree.DragKind Property

Not documented.

Pascal

```
property DragKind;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.38 TVirtualStringTree.DragMode Property

Not documented.

Pascal

```
property DragMode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.39 TVirtualStringTree.DragOperations Property

Read or set which drag operations may be allowed in the tree.

Pascal

```
property DragOperations: TDragOperations;
```

Description

Using this property you can determine, which actions may be performed when a drag operation is finished. The default value includes move, copy and link, where link is rather an esoteric value and only there because it is supported by OLE. The values used directly determine which image is shown for the drag cursor. The specified drag operations do not tell which actions will actually be performed but only, which actions are allowed. They still can be modified during drag'n drop by using a modifier key like the control, shift or alt key or can entirely be ignored by the drop handler.

Class

TVirtualStringTree Class(see page 402)

10.1.16.40 TVirtualStringTree.DragType Property

Read or set which subsystem should be used for dragging.

Pascal

```
property DragType: TVTDragType;
```

Description

Traditionally, Delphi only supports its own drag mechanism, which is not compatible with the rest of the system. This VCL dragging also does not support to transport random data nor does it support drag operations between applications. Thus Virtual Treeview also supports the generally used OLE dragging, which in turn is incompatible with VCL dragging. Depending on your needs you can enable either VCL or OLE dragging as both together cannot be started. However, Virtual Treeview is able to act as drop target for both kind of data, independant of what is set in DragType.

Class

TVirtualStringTree Class(see page 402)

10.1.16.41 TVirtualStringTree.DragWidth Property

Read or set the horizontal limit of the internal drag image.

Pascal

```
property DragWidth: Integer;
```

Description

The DragWidth property (as well as the DragHeight property) are only for compatibility reason in the tree. If a platform does not support the IDropTargetHelper interface (Windows 9x/Me, Windows NT 4.0) then Virtual Treeview uses its own implementation of a DragImage. Since displaying a translucent drag image is performance hungry you should limit the image size shown for the drag operation.

Class

TVirtualStringTree Class(see page 402)

10.1.16.42 TVirtualStringTree.DrawSelectionMode Property

Read or set how multiselection with the mouse is to be visualized.

Pascal

```
property DrawSelectionMode: TVTDrawSelectionMode;
```

Description

Virtuall Treeview allows to display two different selection rectangles when doing multiselection with the mouse. One is the traditiional dotted focus rectangle and the other one is a translucent color rectangle. The latter is the preferred one but the former is set as default (for compatibility reasons).

Class

TVirtualStringTree Class(see page 402)

10.1.16.43 TVirtualStringTree.EditDelay Property

Read or set the maximum time between two single clicks on the same node, which should start node editing.

Pascal

```
property EditDelay: Cardinal;
```

Description

A node edit operation can be started using the keyboard (F2 key), in code using EditNode or by clicking twice on the same node (but not doing a double click). EditDelay is the maxmimum time distance between both clicks in which the edit operation is started.

See Also

Editors and editing(see page 41)

Class

TVirtualStringTree Class(see page 402)

10.1.16.44 TVirtualStringTree.Enabled Property

Not documented.

Pascal

```
property Enabled;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.45 TVirtualStringTree.Font Property

Same as TWinControl.Font.

Pascal

```
property Font;
```

Description

See TWinControl.Font.

Class

TVirtualStringTree Class(see page 402)

10.1.16.46 TVirtualStringTree.Header Property

Provides access to the header instance.

Pascal

```
property Header: TVTHeader;
```

Description

This property is used to allow access to the header instance, which manages all aspects of the tree's header image as well as the column settings.

See Also

TVTHeader

Class

TVirtualStringTree Class(see page 402)

10.1.16.47 TVirtualStringTree.HintAnimation Property

Read or set the current hint animation type.

Pascal

```
property HintAnimation: THintAnimationType;
```

Description

With this property you can specify what animation you would like to play when displaying a hint. For some applications it might not be good to animate hints, hence you can entirely switch them off. Usually however you will leave the system standard. This way the user can decide whether and which hint animation he or she likes.

Class

TVirtualStringTree Class(see page 402)

10.1.16.48 TVirtualStringTree.HintMode Property

Read or set what type of hint you want for the tree view.

Pascal

```
property HintMode: TVTHintMode;
```

Description

Virtual Treeview supports several hints modes. This includes the normal hint used for any other TControl class as well as a node specific hint, which is individual for each node or even each cell.

Class

TVirtualStringTree Class(see page 402)

10.1.16.49 TVirtualStringTree.HotCursor Property

Read or set which cursor should be used for hot nodes.

Pascal

```
property HotCursor: TCursor;
```

Description

When you enable toHotTrack in TreeOptions.PaintOptions then the node, which is currently under the mouse pointer becomes the hot node. This is a special state, which can be used for certain effects. Hot nodes have by default an underlined caption and may cause the cursor to change to what ever you like. The HotCursor property is used to specify, which cursor is to be used.

See Also

HotNode, TVTPaintOptions

Class

TVirtualStringTree Class(see page 402)

10.1.16.50 TVirtualStringTree.Images Property

Read or set the tree's normal image list.

Pascal

```
property Images: TCustomImageList;
```

Description

Just like with TListView and TTreeview also Virtual Treeview can take an image list for its normal images. Additionally, there are image lists for state images and check images.

See Also

StateImages, CheckImages

Class

TVirtualStringTree Class(see page 402)

10.1.16.51 TVirtualStringTree.IncrementalSearch Property

Read or set the current incremental search mode.

Pascal

```
property IncrementalSearch: TVTIncrementalSearch;
```

Description

Virtual Treeview can do an incremental search by calling back the application when comparing node captions. The IncrementalSearch property determines whether incremental search is enabled and which nodes should be searched through.

See Also

IncrementalSearchDirection, IncrementalSearchStart, IncrementalSearchTimeout

Class

TVirtualStringTree Class(see page 402)

10.1.16.52 TVirtualStringTree.IncrementalSearchDirection Property

Read or set the direction to be used for incremental search.

Pascal

```
property IncrementalSearchDirection: TVTSearchDirection;
```

Description

When incremental search is enabled then Virtual Treeview can search forward and backward from the start point given by IncrementalSearchStart.

See Also

IncrementalSearch, IncrementalSearchStart, IncrementalSearchTime123out

Class

TVirtualStringTree Class(see page 402)

10.1.16.53 TVirtualStringTree.IncrementalSearchStart Property

Read or set where to start incremental search.

Pascal

```
property IncrementalSearchStart: TVTSearchStart;
```

Description

When incremental search is enabled in the tree view then you can specify here, where to start the next incremental search operation from.

See Also

IncrementalSearch, IncrementalSearchDirection, IncrementalSearchTimeout

Class

TVirtualStringTree Class(see page 402)

10.1.16.54 TVirtualStringTree.IncrementalSearchTimeout Property

Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Pascal

```
property IncrementalSearchTimeout: Cardinal;
```

Description

When incremental search is enabled in Virtual Treeview then you can specify here after what time incremental search should stop when no keyboard input is encountered any longer. This property so determines also the speed at which users have to type letters to keep the incremental search rolling.

See Also

IncrementalSearch, IncrementalSearchDirection, IncrementalSearchStart

Class

TVirtualStringTree Class(see page 402)

10.1.16.55 TVirtualStringTree.Indent Property

Read or set the indentation amount for node levels.

Pascal

```
property Indent: Cardinal;
```

Description

Each new level in the tree (child nodes of a parent node) are visually shifted to distinguish betwenn them and their parent node (that's the tree layout after all). The Indent property determines the shift distance in pixels.

Class

TVirtualStringTree Class(see page 402)

10.1.16.56 TVirtualStringTree.LineMode Property

Read or set the mode of the tree lines.

Pascal

```
property LineMode: TVTLineMode;
```

Description

Apart from the usual lines Virtual Treeview also supports a special draw mode named bands. This allows for neat visual effects.

Class

TVirtualStringTree Class(see page 402)

10.1.16.57 TVirtualStringTree.LineStyle Property

Read or set the mode of the tree lines.

Pascal

```
property LineStyle: TVTLineStyle;
```

Description

Virtual Treeview allows to customize the lines used to display the node hierarchy. The default style is a dotted pattern, but you can also make solid lines or specify your own line pattern.

Class

TVirtualStringTree Class(see page 402)

10.1.16.58 TVirtualStringTree.Margin Property

Read or set the tree's node margin.

Pascal

```
property Margin: Integer;
```

Description

The node margin is the distance between the cell bounds and its content like the lines, images, check box and so on. However this border is only applied to the left and right side of the node cell.

Note: there is also a TextMargin property in TVirtualStringTree, which is an additional border for the cell text only.

See Also

TVirtualStringTree.TextMargin

Class

TVirtualStringTree Class(see page 402)

10.1.16.59 TVirtualStringTree.NodeAlignment Property

Read or set the node alignment value.

Pascal

```
property NodeAlignment: TVTNodeAlignment;
```

Description

Nodes have got an align member, which is used to determine the vertical position of the node's images and tree lines. The NodeAlignment property specifies how to interpret the value in the align member.

See Also

TVirtualNode

Class

TVirtualStringTree Class(see page 402)

10.1.16.60 TVirtualStringTree.NodeDataSize Property

Read or set the extra data size for each node.

Pascal

```
property NodeDataSize: Integer;
```

Description

A node can have an area for user data, which can be used to store application defined, node specific data in. Use GetNodeData to get the address of this area. In addition to assigning a value here you can also use the OnGetNodeDataSize event, which is called when NodeDataSize is -1.

See Also

Data handling(see page 39)

Class

TVirtualStringTree Class(see page 402)

10.1.16.61 TVirtualStringTree.OnAdvancedHeaderDraw Event

Header paint support event.

Pascal

```
property OnAdvancedHeaderDraw: TVTAdvancedHeaderPaintEvent;
```

Description

The OnAdvancedHeaderDraw event is used when owner draw is enabled for the header and a column is set to owner draw mode. It can be used to custom draw only certain parts of the header instead the whole thing. A good example for this event is customizing the background of the header for only one column. With the standard custom draw method (OnHeaderDraw) you are in an all-or-nothing situation and have to paint everything in the header including the text, images and sort direction indicator. OnAdvancedHeaderDraw however uses OnHeaderDrawQueryElements to ask for the elements the application wants to draw and acts accordingly.

See Also

OnHeaderDrawQueryElements, OnHeaderDraw

Class

TVirtualStringTree Class(see page 402)

10.1.16.62 TVirtualStringTree.OnAfterCellPaint Event

Paint support event.

Pascal

```
property OnAfterCellPaint: TVTAfterCellPaintEvent;
```

Description

This event is called whenever a cell has been painted. A cell is defined as being one part of a node bound to a certain column. This event is called several times per node (the amount is determined by visible columns and size of the part to draw).

See Also

Paint cycles and stages(see page 36)

Class

TVirtualStringTree Class(see page 402)

10.1.16.63 TVirtualStringTree.OnAfterItemErase Event

Paint support event.

Pascal

```
property OnAfterItemErase: TVTAfterItemEraseEvent;
```

Description

Called after the background of a node has been erased (erasing can also be filling with a background image). This event is called once per node in a paint cycle.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualStringTree Class(see page 402)

10.1.16.64 TVirtualStringTree.OnAfterItemPaint Event

Paint support event.

Pascal

```
property OnAfterItemPaint: TVTAfterItemPaintEvent;
```

Description

Called after a node has been drawn. This event is called once per node.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualStringTree Class(see page 402)

10.1.16.65 TVirtualStringTree.OnAfterPaint Event

Paint support event.

Pascal

```
property OnAfterPaint: TVTPaintEvent;
```

Description

Called after all nodes which needed an update have been drawn. This event is called once per paint cycle.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualStringTree Class(see page 402)

10.1.16.66 TVirtualStringTree.OnBeforeCellPaint Event

Paint support event.

Pascal

```
property OnBeforeCellPaint: TVTBeforeCellPaintEvent;
```

Description

This event is called immediately before a cell is painted.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualStringTree Class(see page 402)

10.1.16.67 TVirtualStringTree.OnBeforeItemErase Event

Paint support event.

Pascal

```
property OnBeforeItemErase: TVTBeforeItemEraseEvent;
```

Description

Called when the background of a node is about to be erased.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualStringTree Class(see page 402)

10.1.16.68 TVirtualStringTree.OnBeforeItemPaint Event

Paint support event.

Pascal

```
property OnBeforeItemPaint: TVTBeforeItemPaintEvent;
```

Description

Called after the background of a node has been drawn and just before the node itself is painted. In this event the application gets the opportunity to decide whether a node should be drawn normally or should be skipped. The application can draw the node itself if necessary or leave the node area blank.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualStringTree Class(see page 402)

10.1.16.69 TVirtualStringTree.OnBeforePaint Event

Paint support event.

Pascal

```
property OnBeforePaint: TVTPaintEvent;
```

Description

Called as very first event in a paint cycle. In this event has the application the opportunity to do some special preparation of the canvas onto which the tree is painted, e.g. setting a special viewport and origin or a different mapping mode.

See Also

Paint cycles and stages(see page 36)

Class

TVirtualStringTree Class(see page 402)

10.1.16.70 TVirtualStringTree.OnChange Event

Navigation support event.

Pascal

```
property OnChange: TVTChangeEvent;
```

Description

Called when a node's selection state has changed.

Class

TVirtualStringTree Class(see page 402)

10.1.16.71 TVirtualStringTree.OnChecked Event

Check support event.

Pascal

```
property OnChecked: TVTChangeEvent;
```

Description

Triggered when a node's check state has changed.

Class

TVirtualStringTree Class(see page 402)

10.1.16.72 TVirtualStringTree.OnChecking Event

Check support event.

Pascal

```
property OnChecking: TVTCheckChangingEvent;
```

Description

Triggered when a node's check state is about to change and allows to prevent the change.

Class

TVirtualStringTree Class(see page 402)

10.1.16.73 TVirtualStringTree.OnClick Property

Not documented.

Pascal

```
property OnClick;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.74 TVirtualStringTree.OnCollapsed Event

Miscellaneous event.

Pascal

```
property OnCollapsed: TVTChangeEvent;
```

Description

Triggered after a node has been collapsed, that is, its child nodes are no longer displayed.

Class

TVirtualStringTree Class(see page 402)

10.1.16.75 TVirtualStringTree.OnCollapsing Event

Miscellaneous event.

Pascal

```
property OnCollapsing: TVTChangingEvent;
```

Description

Triggered when a node is about to be collapsed and allows to prevent collapsing the node by setting Allowed to false.

Class

TVirtualStringTree Class(see page 402)

10.1.16.76 TVirtualStringTree.OnColumnClick Event

Header and column support event.

Pascal

```
property OnColumnClick: TVTColumnClickEvent;
```

Description

Triggered when the user released a mouse button over the same column in the client area on which the button was pressed previously.

See Also

OnHeaderClick

Class

TVirtualStringTree Class(see page 402)

10.1.16.77 TVirtualStringTree.OnColumnDblClick Event

Header and column support event.

Pascal

```
property OnColumnDblClick: TVTColumnDblClickEvent;
```

Description

Same as OnColumnClick but for double clicks.

See Also

OnColumnClick, OnHeaderDblClick

Class

TVirtualStringTree Class(see page 402)

10.1.16.78 TVirtualStringTree.OnColumnResize Event

Header and column support routine.

Pascal

```
property OnColumnResize: TVTHeaderNotifyEvent;
```

Description

Triggered when a column is being resized. During resize OnColumnResize is frequently hence you should make any code in the associated event handle a short and fast as possible.

Class

TVirtualStringTree Class(see page 402)

10.1.16.79 TVirtualStringTree.OnCompareNodes Event

Sort and search support event.

Pascal

```
property OnCompareNodes: TVTCompareEvent;
```

Description

This event is the core event for all comparations between nodes. It is important that you write a handler for this event if you want to sort nodes!

Result must be set to less than 0 if Node1 is considered as being before Node2, equal to 0 if both a considered being the same and greater than 0 if the first node is considered as being after node 2. Keep in mind that you don't need to take sort direction into account. This is automatically handled by the tree. Simply return a comparation result as would there be an ascending sort order.

Below is some sample code taken from the Advanced Demo:

```
procedure TMainForm.VDT1CompareNodes(Sender: TBaseVirtualTree; Node1, Node2: PVirtualNode; Column:
Integer;
  var Result: Integer);

// used to sort the image draw tree

var
  Data1,
  Data2: PImageData;

begin
  Data1 := Sender.GetNodeData(Node1);
  Data2 := Sender.GetNodeData(Node2);
  // folder are always before files
  if Data1.IsFolder <> Data2.IsFolder then
  begin
    // one of both is a folder the other a file
    if Data1.IsFolder then
      Result := -1
    else
      Result := 1;
  end
  else // both are of same type (folder or file)
    Result := CompareText(Data1.FullPath, Data2.FullPath);
end;
```

See Also

SortTree, Sort

Class

TVirtualStringTree Class(see page 402)

10.1.16.80 TVirtualStringTree.OnCreateDataObject Event

Drag'n drop support event.

Pascal

```
property OnCreateDataObject: TVTCreateDataObjectEvent;
```

Description

This event is called when the tree's drag manager needs a data object interface to start a drag'n drop operation. Descentants (which override DoGetDataObject) or the application can return an own IDataObject implementation to support special formats.

Class

TVirtualStringTree Class(see page 402)

10.1.16.81 TVirtualStringTree.OnCreateDragManager Event

Drag'n drop support event.

Pascal

```
property OnCreateDragManager: TVTCreateDragManagerEvent;
```

Description

This event is usually not used but allows power users to create their own drag manager to have different actions and/or formats than the internal drag manager.

Class

TVirtualStringTree Class(see page 402)

10.1.16.82 TVirtualStringTree.OnCreateEditor Event

Editing support event.

Pascal

```
property OnCreateEditor: TVTCreateEditorEvent;
```

Description

Allows to supply a customized node editor without changing the tree. TBaseVirtualTree triggers this event and raises an exception if there no editor is returned. If you don't want this then disable edit support for nodes in TreeOptions.MiscOptions. Descentants like TCustomVirtualStringTree supply a generic and simple string editor.

See Also

Editors and editing(see page 41)

Class

TVirtualStringTree Class(see page 402)

10.1.16.83 TVirtualStringTree.OnDblClick Property

Not documented.

Pascal

```
property OnDblClick;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.84 TVirtualStringTree.OnDragAllowed Event

Drag'n drop support event.

Pascal

```
property OnDragAllowed: TVTDragAllowedEvent;
```

Description

This event is called in the mouse button down handler to determine whether the application allows to start a drag operation. Since this check is done in sync with the other code it is much prefered over doing a manual BeginDrag.

Notes

The OnDragAllowed event is called only if the current DragMode is dmManual.

Class

TVirtualStringTree Class(see page 402)

10.1.16.85 TVirtualStringTree.OnDragDrop Event

Drag'n drop support event.

Pascal

```
property OnDragDrop: TVTDragDropEvent;
```

Description

Triggered when either a VCL or a OLE drop action occured. Accepting drag and drop actions is not trivial. In order to maintain a minimum compatibility with the VCL drag'n drop system Virtual Tree accepts not only OLE drop actions but also those issued by the Delphi VCL (which is totally different to the OLE way, unfortunately), provided toAcceptOLEDrop is set in TreeOptions.MiscOptions. The code snippet below is taken from a sample project provided with Virtual Tree. It shows a general way to deal with dropped data. The following check list can be used as orientation and additional comment to the code:

1. Determine what kind of drop data is passed. If DataObject is nil or Formats is empty then the drag source is a VCL control. The event is not triggered for OLE drag'n drop if there is no OLE format is available (which should never occur).
2. If the event is triggered by a VCL control then use Source to access either the control or the drag object, depending on the circumstances of the action.
3. For OLE drag'n drop iterate through the Formats list to find a format you can handle.
4. If you find CF_VIRTUALTREE then the source of the drag operation is a Virtual Treeview. Since this is the native tree format you can pass it to the Sender's ProcessDrop method which will take care to retrieve the data and act depending on Effect and Mode. No further action by the application is usually required in this case.
5. If you do not find CF_VIRTUALTREE then the operation has been initiated by another application, e.g. the Explorer (then you will find CF_HDROP or CF_SHELLIDLIST in formats) or Notepad (then you will get CF_TEXT and perhaps CF_UNICODETEXT) etc., depending on the data which is actually dropped.
6. Use the provided DataObject to get the drop data via IDataObject.GetData and act depending on the format you get.
7. Finally set Effect to either DROPEFFECT_COPY, DROPEFFECT_MOVE or DROPEFFECT_NONE to indicate which operation needs to be finished in Sender when the event returns. If you return DROPEFFECT_MOVE then all marked nodes in the source tree will be deleted, otherwise they stay where they are.

```
procedure TMainForm.VTDragDrop(Sender: TBaseVirtualTree; Source: TObject; DataObject: IDataObject;
  const Formats: array of Word; Shift: TShiftState; Pt: TPoint; var Effect: Integer; Mode: TDropMode);
```

```
var
  I: Integer;
  AttachMode: TVTNodeAttachMode;

begin
  if Length(Formats) > 0 then
  begin
    // OLE drag'n drop
    // If the native tree format is listed then use this and accept the drop, otherwise recject
(ignore) it.
    // It is recommend by Microsoft to order available clipboard formats in decreasing detail richness
so
    // the first best format which we can accept is usually the best format we can get at all.
    for I := 0 to High(Formats) do
      if Formats[I] = CF_VIRTUALTREE then
      begin
        case Mode of
          dmAbove:
            AttachMode := amInsertBefore;
          dmOnNode:
            AttachMode := amAddChildLast;
          dmBelow:
            AttachMode := amInsertAfter;
        else
          if Assigned(Source) and (Source is TBaseVirtualTree) and (Sender <> Source) then
            AttachMode := amInsertBefore
          else
            AttachMode := amNowhere;
        end;
        // in the case the drop target does an optimized move Effect is set to DROPEFFECT_NONE
        // to indicate this also to the drag source (so the source doesn't need to take any further
action)
        Sender.ProcessDrop(DataObject, Sender.DropTargetNode, Effect, AttachMode);
        Break;
      end;
  end
  else
  begin
    // VCL drag'n drop, Effects contains by default both move and copy effect suggestion,
    // as usual the application has to find out what operation is finally to do
    Beep;
  end;
end;
```

Class

TVirtualStringTree Class(see page 402)

10.1.16.86 TVirtualStringTree.OnDragOver Event

Drag'n drop support event.

Pascal

```
property OnDragOver: TVTDragOverEvent;
```

Description

Triggered when Sender is the potential target of a drag'n drop operation. You can use this event to allow or deny a drop operation by setting Allowed to True or False, respectively. For conditions of OLE or VCL drag source see OnDragDrop.

See Also

OnDragDrop

Class

TVirtualStringTree Class(see page 402)

10.1.16.87 TVirtualStringTree.OnEditCancelled Event

Editing support event.

Pascal

```
property OnEditCancelled: TVTEditCancelEvent;
```

Description

Triggered when an edit action has been cancelled.

See Also

Editors and editing(see page 41)

Class

TVirtualStringTree Class(see page 402)

10.1.16.88 TVirtualStringTree.OnEdited Event

Editing support event.

Pascal

```
property OnEdited: TVTEditChangeEvent;
```

Description

Triggered when an edit action has successfully been finished.

See Also

Editors and editing(see page 41)

Class

TVirtualStringTree Class(see page 402)

10.1.16.89 TVirtualStringTree.OnEditing Event

Editing support event.

Pascal

```
property OnEditing: TVTEditChangingEvent;
```

Description

Triggered when a node is about to be edited. Use Allowed to allow or deny this action.

See Also

Editors and editing(see page 41)

Class

TVirtualStringTree Class(see page 402)

10.1.16.90 TVirtualStringTree.OnEndDock Property

Not documented.

Pascal

```
property OnEndDock;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.91 TVirtualStringTree.OnEndDrag Property

Not documented.

Pascal

```
property OnEndDrag;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.92 TVirtualStringTree.OnEnter Property

Not documented.

Pascal

```
property OnEnter;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.93 TVirtualStringTree.OnExit Property

Not documented.

Pascal

```
property OnExit;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.94 TVirtualStringTree.OnExpanded Event

Misscellaneous event.

Pascal

```
property OnExpanded: TVTChangeEvent;
```

Description

Triggered after a node has been expanded.

Class

TVirtualStringTree Class(see page 402)

10.1.16.95 TVirtualStringTree.OnExpanding Event

Miscellaneous event.

Pascal

```
property OnExpanding: TVTChangingEvent;
```

Description

Triggered just before a node is expanded. Use Allowed to allow or deny this action.

Class

TVirtualStringTree Class(see page 402)

10.1.16.96 TVirtualStringTree.OnFocusChanged Event

Navigation support event.

Pascal

```
property OnFocusChanged: TVTFocusChangeEvent;
```

Description

Triggered after the focused node changed. When examining Node keep in mind that it can be nil, meaning there is no focused node.

Class

TVirtualStringTree Class(see page 402)

10.1.16.97 TVirtualStringTree.OnFocusChanging Event

Navigation support event.

Pascal

```
property OnFocusChanging: TVTFocusChangingEvent;
```

Description

Triggered when the node focus is about to change. You can use Allowed to allow or deny a focus change. Keep in mind that either the old or the new node can be nil.

Class

TVirtualStringTree Class(see page 402)

10.1.16.98 TVirtualStringTree.OnFreeNode Event

Data management node.

Pascal

```
property OnFreeNode: TVTFreeNodeEvent;
```

Description

Triggered when a node is about to be freed. This is the ideal place to free/disconnect your own data you associated with Node. Keep in mind, that data which is stored directly in the node does not need to be free by the application. This is part of the node record and will be freed when the node is freed. You should however finalize the data in such a case if it contains references to external memory objects (e.g. variants, strings, interfaces).

Class

TVirtualStringTree Class(see page 402)

10.1.16.99 TVirtualStringTree.OnGetCellIsEmpty Event

Triggered when the tree control needs to know whether a given column is empty.

Pascal

```
property OnGetCellIsEmpty: TVTGetCellIsEmptyEvent;
```

Description

Virtual Treeview supports the concept of column spanning where one cell with too much text to fit into its own space can expand to the right cell neighbors if they are empty. To make this work it is necessary to know if a cell is considered as being empty, whatever this means to an application. The string tree descendant simply checks the text for the given cell and calls back its ancestor if there is no text to further refine if the cell must stay as if it contained something. The ancestor (TBaseVirtualTree) now triggers OnGetCellIsEmpty to let the application decide.

Class

TVirtualStringTree Class(see page 402)

10.1.16.100 TVirtualStringTree.OnGetCursor Event

Miscellaneous event.

Pascal

```
property OnGetCursor: TVTGetCursorEvent;
```

Description

This event is triggered from the WM_SETCURSOR message to allow the application use several individual cursors for a tree. The Cursor property allows to set one cursor for the whole control but not to use separate cursors for different tree parts.

Class

TVirtualStringTree Class(see page 402)

10.1.16.101 TVirtualStringTree.OnGetHeaderCursor Event

Header and column support event.

Pascal

```
property OnGetHeaderCursor: TVTGetHeaderCursorEvent;
```

Description

This event is triggered from the WM_SETCURSOR message to allow the application to define individual cursors for the header part of the tree control.

Class

TVirtualStringTree Class(see page 402)

10.1.16.102 TVirtualStringTree.OnGetHelpContext Event

Miscellaneous event.

Pascal

```
property OnGetHelpContext: TVTHelpContextEvent;
```

Description

This event is usually triggered when the user pressed F1 while the tree has the focus. The tree is iteratively traversed all the way up to the top level parent of the given node until a valid help context index is returned (via this event). When the loop reaches the top level without getting a help index then the tree control's help index is used. If the tree itself does not have a help context index then a further traversal is initiated going up parent by parent of each control in the current window hierarchy until either a valid index is found or there is no more window parent.

Class

TVirtualStringTree Class(see page 402)

10.1.16.103 TVirtualStringTree.OnGetHint Event

Virtual string tree event to query for a custom hint text.

Pascal

```
property OnGetHint: TVSTGetHintEvent;
```

Description

Write an event handler for this event to specify a custom hint for the passed node and column. The TextType will always be ttNormal. This event will only be fired if HintMode is not hmTooltip. The delay for hints can be set as usual: adjust the properties HintPause and HintShortPause of the global Application object.

Class

TVirtualStringTree Class(see page 402)

10.1.16.104 TVirtualStringTree.OnGetImageIndex Event

Display management event.

Pascal

```
property OnGetImageIndex: TVTGetImageEvent;
```

Description

This event is triggered whenever the tree needs the index of an image, be it the normal, the selected or the state image. The event should be as fast as possible because it is at times frequently called when the layout of the node must be

determined, e.g. while doing draw selection with the mouse or painting the tree. Kind determines which image is needed and Column determines for which column of the node the image is needed. This value can be -1 to indicate there is no column used. The parameter Ghosted can be set to true to blend the image 50% against the tree background and can be used for instance in explorer trees to mark hidden file system objects. Additionally nodes are also drawn with a ghosted icon if the are part of a cut set during a pending cut-to-clipboard operation. In this case changing the ghosted parameter has no effect.

Notes

Blending nodes can be switched by using toUseBlendImages in TreeOptions.PaintOptions.

Class

TVirtualStringTree Class(see page 402)

10.1.16.105 TVirtualStringTree.OnGetImageIndexEx Event

Not documented.

Pascal

```
property OnGetImageIndexEx: TVTGetImageExEvent;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.106 TVirtualStringTree.OnGetLineStyle Event

Display management event.

Pascal

```
property OnGetLineStyle: TVTGetLineStyleEvent;
```

Description

This event is used to customize the appearance of the tree and grid lines and is only triggered if the LineStyle property is set to lsCustomStyle. The event must return a pointer to an array containing bits for an 8 x 8 pixel image with word aligned entries. For more info see PrepareBitmaps and the Windows APIs CreateBitmap and CreatePatternBrush.

Notes

It is important that you do not use dynamically allocated memory in this event (also no local variables on the stack). If you do so then either the memory is not valid on return of the event (if allocated on stack) or will never be freed (if allocated with a memory manager). Instead use a constant array and return its address.

See Also

PrepareBitmaps

Class

TVirtualStringTree Class(see page 402)

10.1.16.107 TVirtualStringTree.OnGetNodeDataSize Event

Data management event.

Pascal

```
property OnGetNodeDataSize: TVTGetNodeDataSizeEvent;
```

Description

Triggered when access to a node's data happens the first time but the actual data size is not yet set. Usually you would specify the size of the data you want to have added to each node by NodeDataSize, e.g. SizeOf(TMyRecord) is quite usual there (where TMyRecord is the structure you want to have stored in the node). Sometimes, however it is not possible to determine the node size in advance, so you can leave NodeDataSize being -1 (the default value) and the OnGetNodeDataSize event is triggered as soon as the first regular node is created (the hidden root node does not have user data but internal data which is determined by other means).

See Also

NodeDataSize, Data handling(see page 39)

Class

TVirtualStringTree Class(see page 402)

10.1.16.108 TVirtualStringTree.OnGetPopupMenu Event

Miscellaneous event.

Pascal

```
property OnGetPopupMenu: TVTPopupEvent;
```

Description

This event allows the application to return a popup menu which is specific to a certain node. The tree does an automatic traversal all the way up to the top level node which is the parent of a given node to get a popup menu. If Menu is set then the traversal stops. Otherwise it continues until either a menu is set, AskParent is set to False or the top level parent has been reached.

Class

TVirtualStringTree Class(see page 402)

10.1.16.109 TVirtualStringTree.OnGetText Event

Virtual string tree event to query for a node's normal or static text.

Pascal

```
property OnGetText: TVSTGetTextEvent;
```

Description

This is one of the fundamental string tree events which must always be handled. The string tree will fire this event every time when it needs to know about the text of a specific node and column. This is mainly the case when the node appears in the visible area of the tree view (in other words it is not scrolled out of view) but also on some other occasions, including streaming, drag and drop and calculating the width of the node.

The node text is distinguished between two text types:

- Normal text: If TextType is ttNormal return the main node caption for the specified column.
- Static text: All text that you return when TextType is ttStatic will be displayed right beside the normal text (or left to it if the column's BidiMode is not bdLeftToRight, i.e. the column has right-to-left layout). Static text is used only for informational purposes; it cannot be selected or dragged and if the column is not wide enough to show all text it will not be shortened with an ellipsis (...) as normal text. The string tree will only query for static text if the StringOptions (see TreeOptions) include toShowStaticText. This is off by default.

When this event is fired the text parameter will always be initialized with the value of property DefaultText. To handle the event get your node data and then extract the string for the appropriate column and TextType.

Notes

Be sure that your event handler only contains absolutely necessary code. This event will be fired very often - easily a few hundred times for medium sized trees with some columns defined when the tree is repainted completely.

For example it is far too slow to use Locate() on some Dataset, a database query result or table, and then get the text from some TField. This may only work with in-memory tables or a client dataset. When you initialize your node data do some caching and use these cached values to display the data.

See Also

OnPaintText

Class

TVirtualStringTree Class(see page 402)

10.1.16.110 TVirtualStringTree.OnGetUserClipboardFormats Event

Drag'n drop and clipboard support event.

Pascal

```
property OnGetUserClipboardFormats: TVTGetUserClipboardFormatsEvent;
```

Description

Whenever the tree needs to specify the available clipboard formats for a clipboard or drag'n drop operation it calls this event too, to allow the application or descentants (which would override DoGetUserClipboardFormats) to specify own formats which can be rendered. Since the build-in data object does not know how to render formats which are specified here you have to supply a handler for the OnRenderOLEData event or an own IDataObject implementation to fully support your own formats.

Use the Formats parameter which is an open array and add the identifiers of your formats (which you got when you registered the format).

Class

TVirtualStringTree Class(see page 402)

10.1.16.111 TVirtualStringTree.OnHeaderClick Event

Header & column support event.

Pascal

```
property OnHeaderClick: TVTHeaderClickEvent;
```

Description

This event is triggered when the user clicks on a header button and is usually a good place to set the current SortColumn and SortDirection.

See Also

SortColumn, SortDirection

Class

TVirtualStringTree Class(see page 402)

10.1.16.112 TVirtualStringTree.OnHeaderDblClick Event

Header & column support event.

Pascal

```
property OnHeaderDblClick: TVTHeaderClickEvent;
```

Description

Unlike OnHeaderClick this event is triggered for double clicks on any part of the header and comes with more detailed information like shift state, which mouse button caused the event and the mouse position.

See Also

OnHeaderClick

Class

TVirtualStringTree Class(see page 402)

10.1.16.113 TVirtualStringTree.OnHeaderDragged Event

Header & column support event.

Pascal

```
property OnHeaderDragged: TVTHeaderDraggedEvent;
```

Description

Triggered after the user has released the left mouse button when a header drag operation was active. Column contains the index of the column which was dragged. Use this index for the Columns property of the header to find out the current position. OldPosition is the position which Column occupied before it was dragged around.

Class

TVirtualStringTree Class(see page 402)

10.1.16.114 TVirtualStringTree.OnHeaderDraggedOut Event

Header & column support event.

Pascal

```
property OnHeaderDraggedOut: TVTHeaderDraggedOutEvent;
```

Description

When during a header drag operation the mouse moves out of the header rectangle and the mouse button is released then an OnHeaderDraggedOut event will be fired with the target mouse position in screen coordinates.

Class

TVirtualStringTree Class(see page 402)

10.1.16.115 TVirtualStringTree.OnHeaderDragging Event

Header & column support event.

Pascal

```
property OnHeaderDragging: TVTHeaderDraggingEvent;
```

Description

Triggered just before dragging of a header button starts. Set Allowed to False if you want to prevent the drag operation of the given column.

Class

TVirtualStringTree Class(see page 402)

10.1.16.116 TVirtualStringTree.OnHeaderDraw Event

Header & column support event.

Pascal

```
property OnHeaderDraw: TVTHeaderPaintEvent;
```

Description

If you set the hoOwnerDraw style in TVTHeader.Options and a column has been set to vsOwnerDraw (see also TVirtualTreeColumn.Style) then OnDrawHeader is called whenever a column needs painting.

Class

TVirtualStringTree Class(see page 402)

10.1.16.117 TVirtualStringTree.OnHeaderDrawQueryElements Event

Header & column support event.

Pascal

```
property OnHeaderDrawQueryElements: TVTHeaderPaintQueryElementsEvent;
```

Description

Used for advanced header painting to query the application for the elements, which are drawn by it and which should be drawn by the tree.

See Also

OnAdvancedHeaderDraw

Class

TVirtualStringTree Class(see page 402)

10.1.16.118 TVirtualStringTree.OnHeaderMouseDown Event

Header & column support event.

Pascal

```
property OnHeaderMouseDown: TVTHeaderMouseEvent;
```

Description

This event is similar to OnHeaderClick but comes with more detailed information like shift state, which mouse button caused the event and the mouse position.

Class

TVirtualStringTree Class(see page 402)

10.1.16.119 TVirtualStringTree.OnHeaderMouseMove Event

Header & column support event.

Pascal

```
property OnHeaderMouseMove: TVTHeaderMouseMoveEvent;
```

Description

This event is triggered when the mouse pointer is moved over the header area.

Class

TVirtualStringTree Class(see page 402)

10.1.16.120 TVirtualStringTree.OnHeaderMouseUp Event

Header & column support event.

Pascal

```
property OnHeaderMouseUp: TVTHeaderMouseEvent;
```

Description

This event is very much like OnHeaderMouseDown but is triggered when a mouse button is released.

Class

TVirtualStringTree Class(see page 402)

10.1.16.121 TVirtualStringTree.OnHotChange Event

Navigation support event.

Pascal

```
property OnHotChange: TVTHotNodeChangeEvent;
```

Description

This event is triggered if hot tracking is enabled (see also TreeOptions.PaintOptions) and when the mouse pointer moves from one node caption to another. In full row select mode most parts of a node are considered as being part of the caption.

Class

TVirtualStringTree Class(see page 402)

10.1.16.122 TVirtualStringTree.OnIncrementalSearch Event

Miscellaneous event.

Pascal

```
property OnIncrementalSearch: TVTIncrementalSearchEvent;
```

Description

This event is integral part of the incremental search functionality (see also Keyboard, hotkeys and incremental search). It is triggered during search for a node which matches the given string. Similar to other compare routines return a value < 0 if the node's caption is considered as being before the given text, = 0 if it is the same and > 0 if it is considered being after the given text.

```
procedure TfrmProperties.VST3IncrementalSearch(Sender: TBaseVirtualTree; Node: PVirtualNode; const
Text: WideString;
  var Result: Integer);

var
  S, PropText: string;


begin
  // Note: This code requires a proper Unicode/WideString comparison routine which I did not want to
link here for
  // size and clarity reasons. For now strings are (implicitly) converted to ANSI to make the
comparison work.
  // Search is not case sensitive.
  S := Text;
  if Node.Parent = Sender.RootNode then
  begin
    // root nodes
    if Node.Index = 0 then
      PropText := 'Description'
    else
      PropText := 'Origin';
  end
  else
  begin
    PropText := PropertyTexts[Node.Parent.Index, Node.Index, ptkText];
  end;

  // By using StrLIComp we can specify a maximum length to compare. This allows us to find also nodes
  // which match only partially.
  Result := StrLIComp(PChar(S), PChar(PropText), Min(Length(S), Length(PropText)))
end;
```

Notes

Usually incremental search allows to match also partially. Hence it is recommended to do comparison only up to the length

of the shorter string.

Class

TVirtualStringTree Class(see page 402)

10.1.16.123 TVirtualStringTree.OnInitChildren Event

Node management event.

Pascal

```
property OnInitChildren: TVTInitChildrenEvent;
```

Description

In order to allow the tree only to fill content where needed it is possible to set the vsHasChildren style in a node's initializaton whithout really adding any child nodes. These child nodes must be initialized first when they are about to be displayed or another access (like search, iteration etc.) occurs.

The application usually prepares data needed to fill child nodes when they are initialized and retrieves the actual number. Set ChildCount to the number of children you want.

See Also

The virtual paradigm(see page 33)

Class

TVirtualStringTree Class(see page 402)

10.1.16.124 TVirtualStringTree.OnInitNode Event

Node management event.

Pascal

```
property OnInitNode: TVTInitNodeEvent;
```

Description

This event is important to connect the tree to your internal data. It is the ideal place to put references or whatever you need into a node's data area. You can set some initial states like selection, expansion state or that a node has child nodes.

See Also

The virtual paradigm(see page 33)

Class

TVirtualStringTree Class(see page 402)

10.1.16.125 TVirtualStringTree.OnKeyAction Event

Miscellaneous event.

Pascal

```
property OnKeyAction: TVTKeyActionEvent;
```

Description

This event is a convinient way for the application or descentant trees to change the semantic of a certain key stroke. It is triggered when the user presses a key and allows either to process that key normally (leave DoDefault being True) or

change it to another key instead (set DoDefault to False then). This way a key press can change its meaning or entirely be ignored (if CharCode is set to 0).

Class

TVirtualStringTree Class(see page 402)

10.1.16.126 TVirtualStringTree.OnKeyDown Property

Not documented.

Pascal

```
property OnKeyDown;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.127 TVirtualStringTree.OnKeyPress Property

Not documented.

Pascal

```
property OnKeyPress;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.128 TVirtualStringTree.OnKeyUp Property

Not documented.

Pascal

```
property OnKeyUp;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.129 TVirtualStringTree.OnLoadNode Event

Streaming support event.

Pascal

```
property OnLoadNode: TVTSaveNodeEvent;
```

Description

This event is typically triggered when serialized tree data must be restored, e.g. when loading the tree from file or stream or during a clipboard/drag'n drop operation. You should only read in what you wrote out in OnSaveNode. For safety there is a check in the loader code which tries to keep the internal serialization structure intact in case the application does not read correctly.

See Also

OnSaveNode, LoadFromStream, SaveToStream, AddFromStream, VTTreeStreamVersion, TVTHeader.LoadFromStream, TVTHeader.SaveToStream

Class

TVirtualStringTree Class(see page 402)

10.1.16.130 TVirtualStringTree.OnMeasureItem Event

Miscellaneous event.

Pascal

```
property OnMeasureItem: TVTMeasureItemEvent;
```

Description

Virtual Treeview supports individual node heights. However it might sometimes unpractical to set this height in advance (e.g. during OnInitNode). Another scenario might be that multi line nodes must size themselves to accomodate the entire node text without clipping. For such and similar cases the event OnMeasureItem is for. It is queried once for each node and allows to specify the node's future height. If you later want to have a new height applied (e.g. because the node's text changed) then call InvalidateNode for it and its vsHeightMeasured state is reset causing so the tree to trigger the OnMeasureItem event again when the node is painted the next time.

See Also

InvalidateNode, vsHeightMeasured

Class

TVirtualStringTree Class(see page 402)

10.1.16.131 TVirtualStringTree.OnMouseDown Property

Not documented.

Pascal

```
property OnMouseDown;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.132 TVirtualStringTree.OnMouseMove Property

Not documented.

Pascal

```
property OnMouseMove;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.133 TVirtualStringTree.OnMouseUp Property

Not documented.

Pascal

```
property OnMouseUp;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.134 TVirtualStringTree.OnMouseWheel Property

Not documented.

Pascal

```
property OnMouseWheel;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.135 TVirtualStringTree.OnNewText Event

Virtual string tree event to pass edited text.

Pascal

```
property OnNewText: TVSTNewTextEvent;
```

Description

A string tree will fire this event after a node has been edited successfully (not canceled with Escape). The event handler must store the new text in the node data.

This event will only be used for the default node caption editor. Other custom node editors may or may not use this event to pass their edited data to the application. Editing for the whole tree is only possible if the MiscOptions (see TreeOptions) include toEditable. If only certain columns or nodes should be editable write an event handler for OnEditing.

See Also

OnCreateEditor, OnEdited

Class

TVirtualStringTree Class(see page 402)

10.1.16.136 TVirtualStringTree.OnNodeCopied Event

Miscellaneous event.

Pascal

```
property OnNodeCopied: TVTNodeCopiedEvent;
```

Description

This event is triggered during drag'n drop after a node has been copied to a new location. Sender is the target tree where the copy operation took place.

Class

TVirtualStringTree Class(see page 402)

10.1.16.137 TVirtualStringTree.OnNodeCopying Event

Miscellaneous event.

Pascal

```
property OnNodeCopying: TVTNodeCopyingEvent;
```

Description

This event is triggered when a node is about to be copied to a new location. Use Allowed to allow or deny the action. Sender is the target tree where the copy operation will take place.

Class

TVirtualStringTree Class(see page 402)

10.1.16.138 TVirtualStringTree.OnNodeMoved Event

Miscellaneous event.

Pascal

```
property OnNodeMoved: TVTNodeMovedEvent;
```

Description

This event is very much like OnNodeCopied but used for moving nodes instead.

Class

TVirtualStringTree Class(see page 402)

10.1.16.139 TVirtualStringTree.OnNodeMoving Event

Miscellaneous event.

Pascal

```
property OnNodeMoving: TVTNodeMovingEvent;
```

Description

This event is very much like OnNodeCopying but used for moving nodes instead.

Class

TVirtualStringTree Class(see page 402)

10.1.16.140 TVirtualStringTree.OnPaintBackground Event

Paint support event.

Pascal

```
property OnPaintBackground: TVTBackgroundPaintEvent;
```

Description

This event is triggered when the tree has finished its painting and there is an area which is not covered by nodes. For nodes there are various events to allow background customizaton. For the free area in the tree window there is this event.

Class

TVirtualStringTree Class(see page 402)

10.1.16.141 TVirtualStringTree.OnPaintText Event

Event to change text formatting for particular nodes.

Pascal

```
property OnPaintText: TVTPaintText;
```

Description

Write an event handler for this event to render nodes with different fonts, font sizes, styles or colors. According to the parameters each column of each node and even normal and static text can be painted in different ways.

Notes

The string tree view manages an internal width for each node's main column. This is done because computing this width is quite costly and the width is needed on several occasions. If you change the font which is used to paint a node's text, for example to bold face style, its width changes but the tree view does not know this - it still relies on its cached node width. This may result in cut off selection rectangles among others.

Hence if the width of a node changes after its initialization because it is now formatted differently than before force a recalculation of the node width by calling InvalidateNode (when the conditions for the changed formatting are met - not in the event handler for OnPaintText).

See Also

Paint cycles and stages(see page 36)

Class

TVirtualStringTree Class(see page 402)

10.1.16.142 TVirtualStringTree.OnRenderOLEData Event

Drag'n drop and clipboard support event.

Pascal

```
property OnRenderOLEData: TVTRenderOLEDataEvent;
```

Description

This event is triggered when the data in a clipboard or drag'n drop operation must be rendered but the built-in data object does not know the requested format. This is usually the case when the application (or descentants) have specified their own formats in OnGetUserClipboardFormats.

Class

TVirtualStringTree Class(see page 402)

10.1.16.143 TVirtualStringTree.OnResetNode Event

Node management event.

Pascal

```
property OnResetNode: TVTChangeEvent;
```

Description

For large trees or simply because the content changed it is sometimes necessary to discard a certain node and release all its children. This can be done with ResetNode which will trigger this event.

See Also

ResetNode

Class

TVirtualStringTree Class(see page 402)

10.1.16.144 TVirtualStringTree.OnResize Property

Not documented.

Pascal

```
property OnResize;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.145 TVirtualStringTree.OnSaveNode Event

Streaming support event.

Pascal

```
property OnSaveNode: TVTSaveNodeEvent;
```

Description

This event is triggered whenever a certain node must be serialized into a stream, e.g. for saving to file or for copying to another tree/node during a clipboard or drag'n drop operation. Make sure you only store non-transient data into the stream. Pointers (including long/wide string references) are transient and the application cannot assume to find the data a pointer references on saving at the same place when the node is loaded (see also OnLoadNode). This is even more essential for nodes which are moved or copied between different trees in different processes (applications). Storing strings however is easily done by writing the strings as a whole into the stream.

Notes

For exchanging data between different trees and for general stability improvement I strongly recommend that you insert a kind of identifier as first stream entry when saving a node. This identifier can then be used to determine what data will follow when loading the node later and does normally not required to be stored in the node data.

See Also

OnLoadNode, LoadFromStream, SaveToStream, AddFromStream, VTTreeStreamVersion, TVTHeader.LoadFromStream, TVTHeader.SaveToStream

Class

TVirtualStringTree Class(see page 402)

10.1.16.146 TVirtualStringTree.OnScroll Event

Miscellaneous event.

Pascal

```
property OnScroll: TVTScrollEvent;
```

Description

This event is triggered when the tree is scrolled horizontally or vertically. You can use it to synchronize scrolling of several trees or other controls.

See Also

OffsetXY

Class

TVirtualStringTree Class(see page 402)

10.1.16.147 TVirtualStringTree.OnShortenString Event

String tree event for custom handling of string abbreviations.

Pascal

```
property OnShortenString: TVSTShortenStringEvent;
```

Description

If the text of a node does not fit into its cell (in grid mode) or is too wide for the width of the tree view it is being abbreviated with an ellipsis (...). By default the ellipsis is added to the end of the node text.

Occasionally you may want to shorten the node text at a different position, for example if the node text is a path string and not the last folder or filename should be cut off but rather some mid level folders if possible.

In the handler S must be processed (shortened) and returned in Result. If Done is set to true (default value is false) the

tree view takes over the shortening. This is useful if not all nodes or columns need

Class

TVirtualStringTree Class(see page 402)

10.1.16.148 TVirtualStringTree.OnShowScrollbar Event

Not documented.

Pascal

```
property OnShowScrollbar: TVTScrollbarShowEvent;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.149 TVirtualStringTree.OnStartDock Property

Not documented.

Pascal

```
property OnStartDock;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.150 TVirtualStringTree.OnStartDrag Property

Not documented.

Pascal

```
property OnStartDrag;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.151 TVirtualStringTree.OnStateChange Event

Miscellaneous event.

Pascal

```
property OnStateChange: TVTStateChangeEvent;
```

Description

For special effects or in order to increase performance it is sometimes useful to know when the tree changes one of its internal states like tsIncrementalSearching or tsOLEDragging. The OnStateChange event is triggered each time such a change occurs letting so the application take measures for it.

Class

TVirtualStringTree Class(see page 402)

10.1.16.152 TVirtualStringTree.OnStructureChange Event

Miscellaneous event.

Pascal

```
property OnStructureChange: TVTStructureChangeEvent;
```

Description

This event is triggered when a change in the tree structure is made. That means whenever a node is created or destroyed or a node's child list is change (because a child node was moved, copied etc.) then OnStructureChange is executed.

Class

TVirtualStringTree Class(see page 402)

10.1.16.153 TVirtualStringTree.OnUpdating Event

Miscellaneous event.

Pascal

```
property OnUpdating: TVTUpdatingEvent;
```

Description

This event is triggered when the application or the tree call BeginUpdate or EndUpdate and indicate so when a larger update operation takes place. This can for instance be used to show a hour glass wait cursor.

Class

TVirtualStringTree Class(see page 402)

10.1.16.154 TVirtualStringTree.ParentBiDiMode Property

Not documented.

Pascal

```
property ParentBiDiMode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.155 TVirtualStringTree.ParentColor Property

Not documented.

Pascal

```
property ParentColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.156 TVirtualStringTree.ParentCtl3D Property

Not documented.

Pascal

```
property ParentCtl3D;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.157 TVirtualStringTree.ParentFont Property

Not documented.

Pascal

```
property ParentFont;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.158 TVirtualStringTree.ParentShowHint Property

Not documented.

Pascal

```
property ParentShowHint;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.159 TVirtualStringTree.PopupMenu Property

Not documented.

Pascal

```
property PopupMenu;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.160 TVirtualStringTree.RootNodeCount Property

Read or set the number of nodes on the top level.

Pascal

```
property RootNodeCount: Cardinal;
```

Description

Usually setting RootNodeCount is all what is needed to initially fill the tree. When one of the top level nodes is initialized you can set its ivsHasChildren style. This will then cause to ask to initialize the child nodes. Recursively applied, you can use this principle to create tree nodes on demand (e.g. when their parent is expanded).

Class

TVirtualStringTree Class(see page 402)

10.1.16.161 TVirtualStringTree.ScrollBarOptions Property

Reference to the scroll bar options class.

Pascal

```
property ScrollBarOptions: TScrollBarOptions;
```

Description

Like many other aspects in Virtual Treeview also scrollbars can be customized. See the class itself for further descriptions.

Class

TVirtualStringTree Class(see page 402)

10.1.16.162 TVirtualStringTree.SelectionBlendFactor Property

Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

Pascal

```
property SelectionBlendFactor: Byte;
```

Description

For a visually appealing tree some operations use alpha blending. One of these operations is multi selection using the mouse. Another one is the rectangle drawn around the caption of selected nodes. Both rectangles use the

SelectionBlendFactor to determine how much of the underlying tree image and how much of the rectangles should be seen. The factor can be in the range of [0..255] where 0 means the rectangle is fully transparent and 255 it is fully opaque.

If you don't like to use blended node selection rectangles then switch them off by removing toUseBlendedSelection from TVTPaintOptions. For selecting a certain multi selection rectangle style use DrawSelectionMode.

Notes

Alpha blending is only enabled when the current processor supports MMX instructions. If MMX is not supported then a dotted draw selection rectangle and an opaque node selection rectangle is used.

See Also

DrawSelectionMode, TVTPaintOptions

Class

TVirtualStringTree Class(see page 402)

10.1.16.163 TVirtualStringTree.SelectionCurveRadius Property

Read or set the current corner radius for node selection rectangles.

Pascal

```
property SelectionCurveRadius: Cardinal;
```

Description

This is a special property to determine the radius of the corners of the selection rectangle for a node caption. Virtual Treeview supports not only simple rectangular selection marks but also such with rounded corners. This feature, however, is only available if blended node selection rectangles are disabled.

See Also

SelectionBlendFactor, DrawSelectionMode, TVTPaintOptions

Class

TVirtualStringTree Class(see page 402)

10.1.16.164 TVirtualStringTree.ShowHint Property

Not documented.

Pascal

```
property ShowHint;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.165 TVirtualStringTree.StateImages Property

Reference to the images list which is used for the state images.

Pascal

```
property StateImages: TCustomImageList;
```

Description

Each node can (in each column) have several images. One is the check image which is supplied by internal image lists or a special external list (see also CustomCheckImages). Another one is the state image and yet another one the normal/selected image.

See Also

CheckImages, Images

Class

TVirtualStringTree Class(see page 402)

10.1.16.166 TVirtualStringTree.TabOrder Property

Not documented.

Pascal

```
property TabOrder;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.167 TVirtualStringTree.TabStop Property

Not documented.

Pascal

```
property TabStop;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.168 TVirtualStringTree.TextMargin Property

Read or set the distance of the node caption to its borders.

Pascal

```
property TextMargin: Integer;
```

Description

TextMargin is used to define a border like area within the content rectangle of a node. This rectangle is the area of the node less the space used for indentation, images, lines and node margins and usually contains the text of a node. In order to support finer adjustment there is another margin, which only applies to the left and right border in the content rectangle. This is the text margin.

See Also

Margin

Class

TVirtualStringTree Class(see page 402)

10.1.16.169 TVirtualStringTree.TreeOptions Property

Reference to the tree's options.

Pascal

```
property TreeOptions: TStringTreeOptions;
```

Description

The tree options are one of the main switchs to modify a treeview's behavior. Virtual Treeview supports customizing tree options by descentants. This allows very fine adjustments for derived tree classes, including the decision which properties should be published. For more information about the base options see TCustomVirtualTreeOptions and its descentants.

Class

TVirtualStringTree Class(see page 402)

10.1.16.170 TVirtualStringTree.Visible Property

Not documented.

Pascal

```
property Visible;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class(see page 402)

10.1.16.171 TVirtualStringTree.WantTabs Property

Read or set whether the tree wants to process tabs on its own.

Pascal

```
property WantTabs: Boolean;
```

Description

Usually tab kex strokes advance the input focus from one control to another on a form. For special processing however it is necessary to let the control decide what to do with the given tabulator character. Virtual Treeview needs this character mainly for its grid emulation.

Class

TVirtualStringTree Class(see page 402)

10.1.16.172 TVirtualStringTree.GetOptionsClass Method

Customization helper to determine which options class the tree should use.

Pascal

```
function GetOptionsClass: TTreeOptionsClass; override;
```

Description

GetOptionsClass is a special purpose method to return a certain class which is used by the tree for its options. TVirtualBaseTree always returns TCustomVirtualTreeOptions but descendants can override this method to return own classes.

For ease of use it makes much sense to always use the same name for the tree's options (which is TreeOptions). By using a customized options class, however, the wrong type is returned by this property. Hence it is meaningful to override TreeOptions and return the derived options class. To make this work the tree descendant must additionally provide new access methods for this property. An example can be seen in TVirtualStringTree:

```
  TVirtualStringTree = class(TCustomVirtualStringTree)
  private
    function GetOptions: TStringTreeOptions;
    procedure SetOptions(const Value: TStringTreeOptions);
  protected
    function GetOptionsClass: TTreeOptionsClass; override;
  public
    property Canvas;
  published
    ...
    property TreeOptions: TStringTreeOptions read GetOptions write SetOptions;
    ...
  end;

  ...

//----------------- TVirtualStringTree
--------------------------------------------------------------------------------

function TVirtualStringTree.GetOptions: TStringTreeOptions;

begin
  Result := FOptions as TStringTreeOptions;
end;

//----------------------------------------------------------------------------------------------------
-----------------

procedure TVirtualStringTree.SetOptions(const Value: TStringTreeOptions);

begin
  FOptions.Assign(Value);
end;

//----------------------------------------------------------------------------------------------------
-----------------

function TVirtualStringTree.GetOptionsClass: TTreeOptionsClass;

begin
  Result := TStringTreeOptions;
end;
```

Class

TVirtualStringTree Class(see page 402)

10.1.17 TVirtualTreeColumn Class

Represents a column in a Virtual Treeview.

Pascal

```
TVirtualTreeColumn = class(TCollectionItem);
```

Description

This enhanced collection item, which is organized within the TCollection descentant TVirtualTreeColumns(see TVirtualTreeColumns Class, page 497), manages all aspects of a single column.

Group

Classes(see page 86)

Methods

Assign(see TVirtualTreeColumn.Assign Method , page 491)
Not documented.

ComputeHeaderLayout(see TVirtualTreeColumn.ComputeHeaderLayout Method , page 492)
Calculates the layout of a column header.

Create(see TVirtualTreeColumn.Create Constructor , page 492)
Not documented.

DefineProperties(see TVirtualTreeColumn.DefineProperties Method , page 492)
Not documented.

Destroy(see TVirtualTreeColumn.Destroy Destructor , page 492)
Not documented.

Equals(see TVirtualTreeColumn.Equals Method , page 493)
Not documented.

GetAbsoluteBounds(see TVirtualTreeColumn.GetAbsoluteBounds Method , page 493)
Not documented.

GetDisplayName(see TVirtualTreeColumn.GetDisplayName Method , page 493)
Not documented.

GetOwner(see TVirtualTreeColumn.GetOwner Method , page 493)
Not documented.

GetRect(see TVirtualTreeColumn.GetRect Method , page 494)
Returns the rectangle this column occupies in the header (relative to (0, 0) of the non-client area).

LoadFromStream(see TVirtualTreeColumn.LoadFromStream Method , page 494)
Not documented.

ParentBiDiModeChanged(see TVirtualTreeColumn.ParentBiDiModeChanged Method , page 494)
Not documented.

ParentColorChanged(see TVirtualTreeColumn.ParentColorChanged Method , page 494)
Not documented.

ReadHint(see TVirtualTreeColumn.ReadHint Method , page 495)
Not documented.

ReadText(see TVirtualTreeColumn.ReadText Method , page 495)
Not documented.

RestoreLastWidth(see TVirtualTreeColumn.RestoreLastWidth Method , page 495)
Not documented.

SaveToStream(see TVirtualTreeColumn.SaveToStream Method , page 495)
Not documented.

UseRightToLeftReading(see TVirtualTreeColumn.UseRightToLeftReading Method , page 496)

Not documented.

WriteHint(see TVirtualTreeColumn.WriteHint Method , page 496)

Not documented.

WriteText(see TVirtualTreeColumn.WriteText Method , page 496)

Not documented.

Properties

Alignment(see TVirtualTreeColumn.Alignment Property, page 487)

Not documented.

BiDiMode(see TVirtualTreeColumn.BiDiMode Property, page 487)

Not documented.

Color(see TVirtualTreeColumn.Color Property, page 487)

Not documented.

Hint(see TVirtualTreeColumn.Hint Property, page 488)

Not documented.

ImageIndex(see TVirtualTreeColumn.ImageIndex Property, page 488)

Not documented.

Layout(see TVirtualTreeColumn.Layout Property, page 488)

Not documented.

Left(see TVirtualTreeColumn.Left Property, page 488)

Not documented.

Margin(see TVirtualTreeColumn.Margin Property, page 489)

Not documented.

MaxWidth(see TVirtualTreeColumn.MaxWidth Property, page 489)

Not documented.

MinWidth(see TVirtualTreeColumn.MinWidth Property, page 489)

Not documented.

Options(see TVirtualTreeColumn.Options Property, page 489)

Not documented.

Owner(see TVirtualTreeColumn.Owner Property, page 490)

Not documented.

Position(see TVirtualTreeColumn.Position Property, page 490)

Not documented.

Spacing(see TVirtualTreeColumn.Spacing Property, page 490)

Not documented.

Style(see TVirtualTreeColumn.Style Property, page 490)

Not documented.

Tag(see TVirtualTreeColumn.Tag Property, page 491)

Not documented.

Text(see TVirtualTreeColumn.Text Property, page 491)

Not documented.

Width(see TVirtualTreeColumn.Width Property, page 491)

Not documented.

Legend

published

Property

public

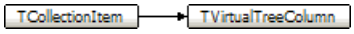
read only

Method

virtual

protected

Class Hierarchy



File

VirtualTrees

10.1.17.1 TVirtualTreeColumn.Alignment Property

Not documented.

Pascal

```
property Alignment: TAlignment;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.2 TVirtualTreeColumn.BiDiMode Property

Not documented.

Pascal

```
property BiDiMode: TBiDiMode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.3 TVirtualTreeColumn.Color Property

Not documented.

Pascal

```
property Color: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.4 TVirtualTreeColumn.Hint Property

Not documented.

Pascal

```
property Hint: WideString;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.5 TVirtualTreeColumn.ImageIndex Property

Not documented.

Pascal

```
property ImageIndex: TImageIndex;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.6 TVirtualTreeColumn.Layout Property

Not documented.

Pascal

```
property Layout: TVTHeaderColumnLayout;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.7 TVirtualTreeColumn.Left Property

Not documented.

Pascal

```
property Left: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.8 TVirtualTreeColumn.Margin Property

Not documented.

Pascal

```
property Margin: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.9 TVirtualTreeColumn.MaxWidth Property

Not documented.

Pascal

```
property MaxWidth: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.10 TVirtualTreeColumn.MinWidth Property

Not documented.

Pascal

```
property MinWidth: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.11 TVirtualTreeColumn.Options Property

Not documented.

Pascal

```
property Options: TVTColumnOptions;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.12 TVirtualTreeColumn.Owner Property

Not documented.

Pascal

```
property Owner: TVirtualTreeColumns;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.13 TVirtualTreeColumn.Position Property

Not documented.

Pascal

```
property Position: TColumnPosition;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.14 TVirtualTreeColumn.Spacing Property

Not documented.

Pascal

```
property Spacing: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.15 TVirtualTreeColumn.Style Property

Not documented.

Pascal

```
property Style: TVirtualTreeColumnStyle;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.16 TVirtualTreeColumn.Tag Property

Not documented.

Pascal

```
property Tag: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.17 TVirtualTreeColumn.Text Property

Not documented.

Pascal

```
property Text: WideString;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.18 TVirtualTreeColumn.Width Property

Not documented.

Pascal

```
property Width: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.19 TVirtualTreeColumn.Assign Method

Not documented.

Pascal

```
procedure Assign(Source: TPersistent); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.20 TVirtualTreeColumn.ComputeHeaderLayout Method

Calculates the layout of a column header.

Pascal

```
procedure ComputeHeaderLayout(DC: HDC; const Client: TRect; UseHeaderGlyph: Boolean; UseSortGlyph:
Boolean; var HeaderGlyphPos: TPoint; var SortGlyphPos: TPoint; var TextBounds: TRect); virtual;
```

Description

The layout of a column header is determined by a lot of factors. This method takes them all into account and determines all necessary positions and bounds:

- for the header text
- the header glyph
- the sort glyph

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.21 TVirtualTreeColumn.Create Constructor

Not documented.

Pascal

```
constructor Create(Collection: TCollection); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.22 TVirtualTreeColumn.DefineProperties Method

Not documented.

Pascal

```
procedure DefineProperties(Filer: TFiler); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.23 TVirtualTreeColumn.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.24 TVirtualTreeColumn.Equals Method

Not documented.

Pascal

```
function Equals(OtherColumn: TVirtualTreeColumn): Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.25 TVirtualTreeColumn.GetAbsoluteBounds Method

Not documented.

Pascal

```
procedure GetAbsoluteBounds(var Left: Integer; var Right: Integer);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.26 TVirtualTreeColumn.GetDisplayName Method

Not documented.

Pascal

```
function GetDisplayName: string; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.27 TVirtualTreeColumn.GetOwner Method

Not documented.

Pascal

```
function GetOwner: TVirtualTreeColumns; reintroduce;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.28 TVirtualTreeColumn.GetRect Method

Returns the rectangle this column occupies in the header (relative to (0, 0) of the non-client area).

Pascal

```
function GetRect: TRect; virtual;
```

Description

Returns the rectangle this column occupies in the header (relative to (0, 0) of the non-client area).

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.29 TVirtualTreeColumn.LoadFromStream Method

Not documented.

Pascal

```
procedure LoadFromStream(const Stream: TStream; Version: Integer);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.30 TVirtualTreeColumn.ParentBiDiModeChanged Method

Not documented.

Pascal

```
procedure ParentBiDiModeChanged;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.31 TVirtualTreeColumn.ParentColorChanged Method

Not documented.

Pascal

```
procedure ParentColorChanged;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.32 TVirtualTreeColumn.ReadHint Method

Not documented.

Pascal

```
procedure ReadHint(Reader: TReader);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.33 TVirtualTreeColumn.ReadText Method

Not documented.

Pascal

```
procedure ReadText(Reader: TReader);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.34 TVirtualTreeColumn.RestoreLastWidth Method

Not documented.

Pascal

```
procedure RestoreLastWidth;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.35 TVirtualTreeColumn.SaveToStream Method

Not documented.

Pascal

```
procedure SaveToStream(const Stream: TStream);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.36 TVirtualTreeColumn.UseRightToLeftReading Method

Not documented.

Pascal

```
function UseRightToLeftReading: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.37 TVirtualTreeColumn.WriteHint Method

Not documented.

Pascal

```
procedure WriteHint(Writer: TWriter);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.17.38 TVirtualTreeColumn.WriteText Method

Not documented.

Pascal

```
procedure WriteText(Writer: TWriter);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class(see page 485)

10.1.18 TVirtualTreeColumns Class

Collection class, which holds the columns for the tree.

Pascal

```
TVirtualTreeColumns = class(TCollection);
```

Description

This class is an enhanced collection which manages general aspects of columns like ordering, traversion, streaming, painting, dragging etc.

Group

Classes(see page 86)

Methods

Add(see TVirtualTreeColumns.Add Method , page 500)
Not documented.

AdjustAutoSize(see TVirtualTreeColumns.AdjustAutoSize Method , page 501)
Called when columns must be sized so that the fit the client area.

AdjustDownColumn(see TVirtualTreeColumns.AdjustDownColumn Method , page 501)
Determines the column from the given position and returns it.

AdjustHoverColumn(see TVirtualTreeColumns.AdjustHoverColumn Method , page 501)
Determines the new hover column index and returns true if the index actually changed else False.

AdjustPosition(see TVirtualTreeColumns.AdjustPosition Method , page 502)
Reorders the column position array so that the given column gets the given position.

AnimatedResize(see TVirtualTreeColumns.AnimatedResize Method , page 502)
Resizes the given column animated by scrolling the window DC.

Assign(see TVirtualTreeColumns.Assign Method , page 502)
Not documented.

Clear(see TVirtualTreeColumns.Clear Method , page 502)
Not documented.

ColumnFromPosition(see TVirtualTreeColumns.ColumnFromPosition Method (TColumnPosition), page 503)
Returns the index of the column at the given position.

Create(see TVirtualTreeColumns.Create Constructor , page 503)
Not documented.

Destroy(see TVirtualTreeColumns.Destroy Destructor , page 503)
Not documented.

DrawButtonText(see TVirtualTreeColumns.DrawButtonText Method , page 504)
Not documented.

DrawXPButton(see TVirtualTreeColumns.DrawXPButton Method , page 504)
Helper procedure to draw an Windows XP like header button.

Equals(see TVirtualTreeColumns.Equals Method , page 504)
Compares itself with the given set of columns.

FixPositions(see TVirtualTreeColumns.FixPositions Method , page 504)
Fixes column positions after loading from DFM.

GetColumnAndBounds(see TVirtualTreeColumns.GetColumnAndBounds Method , page 505)
Returns the column where the mouse is currently in as well as the left and right bound of this column.

GetColumnBounds(see TVirtualTreeColumns.GetColumnBounds Method , page 505)
Returns the left and right bound of the given column.

GetFirstVisibleColumn(see TVirtualTreeColumns.GetFirstVisibleColumn Method , page 505)

Returns the index of the first visible column or "InvalidColumn(see InvalidColumn Constant, page 685)" if either no columns are defined or all columns are hidden.

GetLastVisibleColumn(see TVirtualTreeColumns.GetLastVisibleColumn Method , page 506)

Returns the index of the last visible column or "InvalidColumn(see InvalidColumn Constant, page 685)" if either no columns are defined or all columns are hidden.

GetNextColumn(see TVirtualTreeColumns.GetNextColumn Method , page 506)

Returns the next column in display order. Column is the index of an item in the collection (a column).

GetNextVisibleColumn(see TVirtualTreeColumns.GetNextVisibleColumn Method , page 506)

Returns the next visible column in display order, Column is an index into the columns list.

GetOwner(see TVirtualTreeColumns.GetOwner Method , page 506)

Not documented.

GetPreviousColumn(see TVirtualTreeColumns.GetPreviousColumn Method , page 507)

Returns the previous column in display order, Column is an index into the columns list.

GetPreviousVisibleColumn(see TVirtualTreeColumns.GetPreviousVisibleColumn Method , page 507)

Returns the previous column in display order, Column is an index into the columns list.

GetVisibleColumns(see TVirtualTreeColumns.GetVisibleColumns Method , page 507)

Returns a list of all currently visible columns in actual order.

GetVisibleFixedWidth(see TVirtualTreeColumns.GetVisibleFixedWidth Method , page 507)

Not documented.

HandleClick(see TVirtualTreeColumns.HandleClick Method , page 508)

Generates a click event if the mouse button has been released over the same column it was pressed first.

IndexChanged(see TVirtualTreeColumns.IndexChanged Method , page 508)

Called by a column when its index in the collection changes.

InitializePositionArray(see TVirtualTreeColumns.InitializePositionArray Method , page 508)

Ensures that the column position array contains as much entries as columns are defined.

IsValidColumn(see TVirtualTreeColumns.IsValidColumn Method , page 508)

Determines whether the given column is valid or not, that is, whether it is one of the current columns.

LoadFromStream(see TVirtualTreeColumns.LoadFromStream Method , page 509)

Not documented.

PaintHeader(see TVirtualTreeColumns.PaintHeader Method , page 509)

Not documented.

SaveToStream(see TVirtualTreeColumns.SaveToStream Method , page 509)

Not documented.

TotalWidth(see TVirtualTreeColumns.TotalWidth Method , page 509)

Not documented.

Update(see TVirtualTreeColumns.Update Method , page 510)

Not documented.

UpdatePositions(see TVirtualTreeColumns.UpdatePositions Method , page 510)

Recalculates the left border of every column and updates their position property according to the PostionToIndex array, which primarily determines where each column is placed visually.

Properties

ClickIndex(see TVirtualTreeColumns.ClickIndex Property, page 499)

Not documented.

Header(see TVirtualTreeColumns.Header Property, page 499)

Not documented.

HeaderBitmap(see TVirtualTreeColumns.HeaderBitmap Property, page 499)

Not documented.

Items(see TVirtualTreeColumns.Items Property, page 500)

Not documented.

PositionToIndex(see TVirtualTreeColumns.PositionToIndex Property, page 500)

Not documented.

TrackIndex(see TVirtualTreeColumns.TrackIndex Property, page 500)

Not documented.

Legend

public

Property

read only

protected

Method

virtual

Class Hierarchy

File

VirtualTrees

10.1.18.1 TVirtualTreeColumns.ClickIndex Property

Not documented.

Pascal

```
property ClickIndex: TColumnIndex;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.2 TVirtualTreeColumns.Header Property

Not documented.

Pascal

```
property Header: TVTHeader;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.3 TVirtualTreeColumns.HeaderBitmap Property

Not documented.

Pascal

```
property HeaderBitmap: TBitmap;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.4 TVirtualTreeColumns.Items Property

Not documented.

Pascal

```
property Items [Index: TColumnIndex]: TVirtualTreeColumn;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.5 TVirtualTreeColumns.PositionToIndex Property

Not documented.

Pascal

```
property PositionToIndex: TIndexArray;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.6 TVirtualTreeColumns.TrackIndex Property

Not documented.

Pascal

```
property TrackIndex: TColumnIndex;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.7 TVirtualTreeColumns.Add Method

Not documented.

Pascal

```
function Add: TVirtualTreeColumn; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.8 TVirtualTreeColumns.AdjustAutoSize Method

Called when columns must be sized so that the fit the client area.

Pascal

```
procedure AdjustAutoSize(CurrentIndex: TColumnIndex; Force: Boolean = False);
```

Description

Called only if the header is in auto-size mode which means a column needs to be so large that it fills all the horizontal space not occupied by the other columns. CurrentIndex (if not InvalidColumn(see InvalidColumn Constant, page 685)) describes which column has just been resized.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.9 TVirtualTreeColumns.AdjustDownColumn Method

Determines the column from the given position and returns it.

Pascal

```
function AdjustDownColumn(P: TPoint): TColumnIndex;
```

Description

If this column is allowed to be clicked then it is also kept for later use.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.10 TVirtualTreeColumns.AdjustHoverColumn Method

Determines the new hover column index and returns true if the index actually changed else False.

Pascal

```
function AdjustHoverColumn(P: TPoint): Boolean;
```

Description

Determines the new hover column index and returns true if the index actually changed else False.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.11 TVirtualTreeColumns.AdjustPosition Method

Reorders the column position array so that the given column gets the given position.

Pascal

```
procedure AdjustPosition(Column: TVirtualTreeColumn; Position: Cardinal);
```

Description

Reorders the column position array so that the given column gets the given position.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.12 TVirtualTreeColumns.AnimatedResize Method

Resizes the given column animated by scrolling the window DC.

Pascal

```
procedure AnimatedResize(Column: TColumnIndex; NewWidth: Integer);
```

Description

Resizes the given column animated by scrolling the window DC.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.13 TVirtualTreeColumns.Assign Method

Not documented.

Pascal

```
procedure Assign(Source: TPersistent); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.14 TVirtualTreeColumns.Clear Method

Not documented.

Pascal

```
procedure Clear; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.15 ColumnFromPosition

10.1.18.15.1 TVirtualTreeColumns.ColumnFromPosition Method (TColumnPosition)

Returns the index of the column at the given position.

Pascal

```
function ColumnFromPosition(PositionIndex: TColumnPosition): TColumnIndex; virtual; overload;
```

Description

Returns the index of the column at the given position.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.15.2 TVirtualTreeColumns.ColumnFromPosition Method (TPoint, Boolean)

Determines the current column based on the position passed in P.

Pascal

```
function ColumnFromPosition(P: TPoint; Relative: Boolean = True): TColumnIndex; virtual; overload;
```

Description

Determines the current column based on the position passed in P.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.16 TVirtualTreeColumns.Create Constructor

Not documented.

Pascal

```
constructor Create(AOwner: TVTHeader);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.17 TVirtualTreeColumns.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.18 TVirtualTreeColumns.DrawButtonText Method

Not documented.

Pascal

```
procedure DrawButtonText(DC: HDC; Caption: WideString; Bounds: TRect; Enabled: Boolean; Hot: Boolean;
DrawFormat: Cardinal);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.19 TVirtualTreeColumns.DrawXPButton Method

Helper procedure to draw an Windows XP like header button.

Pascal

```
procedure DrawXPButton(DC: HDC; ButtonR: TRect; DrawSplitter: Boolean; Down: Boolean; Hover: Boolean);
```

Description

Helper procedure to draw an Windows XP like header button.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.20 TVirtualTreeColumns.Equals Method

Compares itself with the given set of columns.

Pascal

```
function Equals(OtherColumns: TVirtualTreeColumns): Boolean;
```

Description

Equals returns true if all published properties are the same (including column order), otherwise false is returned.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.21 TVirtualTreeColumns.FixPositions Method

Fixes column positions after loading from DFM.

Pascal

```
procedure FixPositions;
```

Description

Fixes column positions after loading from DFM.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.22 TVirtualTreeColumns.GetColumnAndBounds Method

Returns the column where the mouse is currently in as well as the left and right bound of this column.

Pascal

```
function GetColumnAndBounds(P: TPoint; var ColumnLeft: Integer; var ColumnRight: Integer; Relative:
Boolean = True): Integer;
```

Description

Left and Right are undetermined if no column is involved.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.23 TVirtualTreeColumns.GetColumnBounds Method

Returns the left and right bound of the given column.

Pascal

```
procedure GetColumnBounds(Column: TColumnIndex; var Left: Integer; var Right: Integer);
```

Description

If Column is NoColumn(see NoColumn Constant, page 686) then the entire client width is returned.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.24 TVirtualTreeColumns.GetFirstVisibleColumn Method

Returns the index of the first visible column or "InvalidColumn(see InvalidColumn Constant, page 685)" if either no columns are defined or all columns are hidden.

Pascal

```
function GetFirstVisibleColumn: TColumnIndex;
```

Description

Returns the index of the first visible column or "InvalidColumn(see InvalidColumn Constant, page 685)" if either no columns are defined or all columns are hidden.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.25 TVirtualTreeColumns.GetLastVisibleColumn Method

Returns the index of the last visible column or "InvalidColumn(see InvalidColumn Constant, page 685)" if either no columns are defined or all columns are hidden.

Pascal

```
function GetLastVisibleColumn: TColumnIndex;
```

Description

Returns the index of the last visible column or "InvalidColumn(see InvalidColumn Constant, page 685)" if either no columns are defined or all columns are hidden.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.26 TVirtualTreeColumns.GetNextColumn Method

Returns the next column in display order. Column is the index of an item in the collection (a column).

Pascal

```
function GetNextColumn(Column: TColumnIndex): TColumnIndex;
```

Description

Returns the next column in display order. Column is the index of an item in the collection (a column).

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.27 TVirtualTreeColumns.GetNextVisibleColumn Method

Returns the next visible column in display order, Column is an index into the columns list.

Pascal

```
function GetNextVisibleColumn(Column: TColumnIndex): TColumnIndex;
```

Description

Returns the next visible column in display order, Column is an index into the columns list.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.28 TVirtualTreeColumns.GetOwner Method

Not documented.

Pascal

```
function GetOwner: TPersistent; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.29 TVirtualTreeColumns.GetPreviousColumn Method

Returns the previous column in display order, Column is an index into the columns list.

Pascal

```
function GetPreviousColumn(Column: TColumnIndex): TColumnIndex;
```

Description

Returns the previous column in display order, Column is an index into the columns list.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.30 TVirtualTreeColumns.GetPreviousVisibleColumn Method

Returns the previous column in display order, Column is an index into the columns list.

Pascal

```
function GetPreviousVisibleColumn(Column: TColumnIndex): TColumnIndex;
```

Description

Returns the previous column in display order, Column is an index into the columns list.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.31 TVirtualTreeColumns.GetVisibleColumns Method

Returns a list of all currently visible columns in actual order.

Pascal

```
function GetVisibleColumns: TColumnsArray;
```

Description

Returns a list of all currently visible columns in actual order.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.32 TVirtualTreeColumns.GetVisibleFixedWidth Method

Not documented.

Pascal

```
function GetVisibleFixedWidth: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.33 TVirtualTreeColumns.HandleClick Method

Generates a click event if the mouse button has been released over the same column it was pressed first.

Pascal

```
procedure HandleClick(P: TPoint; Button: TMouseButton; Force: Boolean; DblClick: Boolean);
```

Description

Alternatively, Force might be set to true to indicate that the down index does not matter (right, middle and double click).

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.34 TVirtualTreeColumns.IndexChanged Method

Called by a column when its index in the collection changes.

Pascal

```
procedure IndexChanged(OldIndex: Integer; NewIndex: Integer);
```

Description

If NewIndex is -1 then the column is about to be removed otherwise it is moved to a new index. The method will then update(see TVirtualTreeColumns.Update Method , page 510) the position array to reflect the change.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.35 TVirtualTreeColumns.InitializePositionArray Method

Ensures that the column position array contains as much entries as columns are defined.

Pascal

```
procedure InitializePositionArray;
```

Description

The array is resized and initialized with default values if needed.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.36 TVirtualTreeColumns.IsValidColumn Method

Determines whether the given column is valid or not, that is, whether it is one of the current columns.

Pascal

```
function IsValidColumn(Column: TColumnIndex): Boolean;
```

Description

Determines whether the given column is valid or not, that is, whether it is one of the current columns.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.37 TVirtualTreeColumns.LoadFromStream Method

Not documented.

Pascal

```
procedure LoadFromStream(const Stream: TStream; Version: Integer);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.38 TVirtualTreeColumns.PaintHeader Method

Not documented.

Pascal

```
procedure PaintHeader(DC: HDC; R: TRect; HOffset: Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.39 TVirtualTreeColumns.SaveToStream Method

Not documented.

Pascal

```
procedure SaveToStream(const Stream: TStream);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.40 TVirtualTreeColumns.TotalWidth Method

Not documented.

Pascal

```
function TotalWidth: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.41 TVirtualTreeColumns.Update Method

Not documented.

Pascal

```
procedure Update(Item: TCollectionItem); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class(see page 497)

10.1.18.42 TVirtualTreeColumns.UpdatePositions Method

Recalculates the left border of every column and updates their position property according to the PostionToIndex array, which primarily determines where each column is placed visually.

Pascal

```
procedure UpdatePositions(Force: Boolean = False);
```

Class

TVirtualTreeColumns Class(see page 497)

10.1.19 TVirtualTreeHintWindow Class

Internally used hint window class to support Unicode hints.

Pascal

```
TVirtualTreeHintWindow = class(THintWindow);
```

Description

TVirtualTreeHintWindow replaces Delphi's own hint window, but only for the tree controls. For the rest of the application the hint stays at it is. This means not the global HintWindowClass variable is changed but only the locally used class by properly responding to CM_HINTSHOW.

Group

Classes(see page 86)

Methods

ActivateHint(see TVirtualTreeHintWindow.ActivateHint Method , page 511)
Not documented.

CalcHintRect(see TVirtualTreeHintWindow.CalcHintRect Method , page 511)
Not documented.

Create(see TVirtualTreeHintWindow.Create Constructor , page 512)
Not documented.

CreateParams(see TVirtualTreeHintWindow.CreateParams Method , page 512)
Not documented.

Destroy(see TVirtualTreeHintWindow.Destroy Destructor , page 512)
Not documented.

IsHintMsg(see TVirtualTreeHintWindow.IsHintMsg Method , page 512)
The VCL is a bit too generous when telling that an existing hint can be cancelled.

Paint(see TVirtualTreeHintWindow.Paint Method , page 513)
Not documented.

Legend

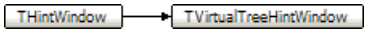
public

Method

virtual

protected

Class Hierarchy



File

VirtualTrees

10.1.19.1 TVirtualTreeHintWindow.ActivateHint Method

Not documented.

Pascal

```
procedure ActivateHint(Rect: TRect; const AHint: string); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeHintWindow Class(see page 510)

10.1.19.2 TVirtualTreeHintWindow.CalcHintRect Method

Not documented.

Pascal

```
function CalcHintRect(MaxWidth: Integer; const AHint: string; AData: Pointer): TRect; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeHintWindow Class(see page 510)

10.1.19.3 TVirtualTreeHintWindow.Create Constructor

Not documented.

Pascal

```
constructor Create(AOwner: TComponent); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeHintWindow Class(see page 510)

10.1.19.4 TVirtualTreeHintWindow.CreateParams Method

Not documented.

Pascal

```
procedure CreateParams(var Params: TCreateParams); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeHintWindow Class(see page 510)

10.1.19.5 TVirtualTreeHintWindow.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeHintWindow Class(see page 510)

10.1.19.6 TVirtualTreeHintWindow.IsHintMsg Method

The VCL is a bit too generous when telling that an existing hint can be cancelled.

Pascal

```
function IsHintMsg(var Msg: TMsg): Boolean; override;
```

Description

Need to specify further here.

Class

TVirtualTreeHintWindow Class(see page 510)

10.1.19.7 TVirtualTreeHintWindow.Paint Method

Not documented.

Pascal

```
procedure Paint; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeHintWindow Class(see page 510)

10.1.20 TVirtualTreeOptions Class

Collects all binary options of the tree control into one place for easier access.

Pascal

```
TVirtualTreeOptions = class(TCustomVirtualTreeOptions);
```

Description

TVirtualTreeOptions does not add any new functionality to TCustomVirtualTreeOptions(see TCustomVirtualTreeOptions Class, page 308) but is the publicly available class.

Group

Classes(see page 86)

Methods

TCustomVirtualTreeOptions Class

AssignTo(see TCustomVirtualTreeOptions.AssignTo Method , page 310)
Used to copy this option class to another option collection.

Create(see TCustomVirtualTreeOptions.Create Constructor , page 310)
Constructor of the class.

Properties

AnimationOptions(see TVirtualTreeOptions.AnimationOptions Property, page 514)
Options related to animations.

AutoOptions(see TVirtualTreeOptions.AutoOptions Property, page 514)
Options related to automatic actions.

MiscOptions(see TVirtualTreeOptions.MiscOptions Property, page 515)
Options not related to any other category.

PaintOptions(see TVirtualTreeOptions.PaintOptions Property, page 515)
Options related to painting.

SelectionOptions(see TVirtualTreeOptions.SelectionOptions Property, page 515)
Options related to the way nodes can be selected.

TCustomVirtualTreeOptions Class

AnimationOptions(see TCustomVirtualTreeOptions.AnimationOptions Property, page 309)
Options related to animations.

AutoOptions(see TCustomVirtualTreeOptions.AutoOptions Property, page 309)
Options related to automatic actions.

MiscOptions(see TCustomVirtualTreeOptions.MiscOptions Property, page 309)
Options not related to any other category.

Owner(see TCustomVirtualTreeOptions.Owner Property, page 309)
Owner tree to which the property class belongs.

PaintOptions(see TCustomVirtualTreeOptions.PaintOptions Property, page 310)
Options related to painting.

SelectionOptions(see TCustomVirtualTreeOptions.SelectionOptions Property, page 310)
Options related to the way nodes can be selected.

Legend

published

Property

protected

public

read only

Method

virtual

Class Hierarchy

File

VirtualTrees

10.1.20.1 TVirtualTreeOptions.AnimationOptions Property

Options related to animations.

Pascal

```
property AnimationOptions: TVTAnimationOptions;
```

Description

These options can be used to switch certain animation effects in a tree.

Class

TVirtualTreeOptions Class(see page 513)

10.1.20.2 TVirtualTreeOptions.AutoOptions Property

Options related to automatic actions.

Pascal

```
property AutoOptions: TVTAutoOptions;
```

Description

These options can be used to switch certain actions in a tree which happen automatically under certain circumstances.

Class

TVirtualTreeOptions Class(see page 513)

10.1.20.3 TVirtualTreeOptions.MiscOptions Property

Options not related to any other category.

Pascal

```
property MiscOptions: TVTMiscOptions;
```

Description

These options can be used to switch miscellanous aspects in a tree.

Class

TVirtualTreeOptions Class(see page 513)

10.1.20.4 TVirtualTreeOptions.PaintOptions Property

Options related to painting.

Pascal

```
property PaintOptions: TVTPaintOptions;
```

Description

These options can be used to switch visual aspects of a tree.

Class

TVirtualTreeOptions Class(see page 513)

10.1.20.5 TVirtualTreeOptions.SelectionOptions Property

Options related to the way nodes can be selected.

Pascal

```
property SelectionOptions: TVTSelectionOptions;
```

Description

These options can be used to switch the way how nodes can be selected in a tree.

Class

TVirtualTreeOptions Class(see page 513)

10.1.21 TVTColors Class

Collects all color related options for the tree control.

Pascal

```
TVTColors = class(TPersistent);
```

Description

TVTColors makes it much more conventient to adjust Virtual Treeview's colors. Since everything is in one place you can also easily compare all colors.

Group

Classes(see page 86)

Methods

Assign(see TVTColors.Assign Method , page 521)
Not documented.

Create(see TVTColors.Create Constructor , page 521)
Not documented.

Properties

BorderColor(see TVTColors.BorderColor Property, page 517)
Not documented.

DisabledColor(see TVTColors.DisabledColor Property, page 517)
Not documented.

DropMarkColor(see TVTColors.DropMarkColor Property, page 517)
Color of the drop mark.

DropTargetBorderColor(see TVTColors.DropTargetBorderColor Property, page 518)
Not documented.

DropTargetColor(see TVTColors.DropTargetColor Property, page 518)
Not documented.

FocusedSelectionBorderColor(see TVTColors.FocusedSelectionBorderColor Property, page 518)
Not documented.

FocusedSelectionColor(see TVTColors.FocusedSelectionColor Property, page 518)
Not documented.

GridLineColor(see TVTColors.GridLineColor Property, page 519)
Not documented.

HeaderHotColor(see TVTColors.HeaderHotColor Property, page 519)
Not documented.

HotColor(see TVTColors.HotColor Property, page 519)
Not documented.

SelectionRectangleBlendColor(see TVTColors.SelectionRectangleBlendColor Property, page 519)
Not documented.

SelectionRectangleBorderColor(see TVTColors.SelectionRectangleBorderColor Property, page 520)
Not documented.

TreeLineColor(see TVTColors.TreeLineColor Property, page 520)
Not documented.

UnfocusedSelectionBorderColor(see TVTColors.UnfocusedSelectionBorderColor Property, page 520)
Not documented.

UnfocusedSelectionColor(see TVTColors.UnfocusedSelectionColor Property, page 520)
Not documented.

Legend

published

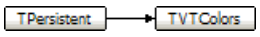
Property

public

Method

virtual

Class Hierarchy



File

VirtualTrees

10.1.21.1 TVTColors.BorderColor Property

Not documented.

Pascal

```
property BorderColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.2 TVTColors.DisabledColor Property

Not documented.

Pascal

```
property DisabledColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.3 TVTColors.DropMarkColor Property

Color of the drop mark.

Pascal

```
property DropMarkColor: TColor;
```

Description

Since the drop metapher has been extended to include dropping on node, above a node or below a node (e.g. to determine adding as child, previous sibling or next sibling) there must be an indication where the node would actually be placed when it would be dropped. This indication is the drop mark, whose color can be set via the DropMarkColor property.

Class

TVTColors Class(see page 515)

10.1.21.4 TVTColors.DropTargetBorderColor Property

Not documented.

Pascal

```
property DropTargetBorderColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.5 TVTColors.DropTargetColor Property

Not documented.

Pascal

```
property DropTargetColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.6 TVTColors.FocusedSelectionBorderColor Property

Not documented.

Pascal

```
property FocusedSelectionBorderColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.7 TVTColors.FocusedSelectionColor Property

Not documented.

Pascal

```
property FocusedSelectionColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.8 TVTColors.GridLineColor Property

Not documented.

Pascal

```
property GridLineColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.9 TVTColors.HeaderHotColor Property

Not documented.

Pascal

```
property HeaderHotColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.10 TVTColors.HotColor Property

Not documented.

Pascal

```
property HotColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.11 TVTColors.SelectionRectangleBlendColor Property

Not documented.

Pascal

```
property SelectionRectangleBlendColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.12 TVTColors.SelectionRectangleBorderColor Property

Not documented.

Pascal

```
property SelectionRectangleBorderColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.13 TVTColors.TreeLineColor Property

Not documented.

Pascal

```
property TreeLineColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.14 TVTColors.UnfocusedSelectionBorderColor Property

Not documented.

Pascal

```
property UnfocusedSelectionBorderColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.15 TVTColors.UnfocusedSelectionColor Property

Not documented.

Pascal

```
property UnfocusedSelectionColor: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.16 TVTColors.Assign Method

Not documented.

Pascal

```
procedure Assign(Source: TPersistent); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.21.17 TVTColors.Create Constructor

Not documented.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTColors Class(see page 515)

10.1.22 TVTDataObject Class

Implementation of an IDataObject interface.

Pascal

```
TVTDataObject = class(TInterfacedObject, IDataObject);
```

Description

This class is used for OLE drag'n drop and clipboard operations. It allows not only to transfer various kinds of data between trees but also to transfer this data between different processes. Additionally, every OLE aware application (like Word) can take part in the data transfer. This makes it easy to copy some of the tree's content for documentation purposes.

Group

Classes(see page 86)

Methods

CanonicalIUnknown(see TVTDataObject.CanonicalIUnknown Method , page 524)
Helper method for setting data in the IDataObject.

Create(see TVTDataObject.Create Constructor , page 524)
Constructor of the class.

DAdvise(see TVTDataObject.DAdvise Method , page 524)
Implementation of the IDataObject.DAdvise method.

Destroy(see TVTDataObject.Destroy Destructor , page 525)
Destructor of the class.

DUnadvise(see TVTDataObject.DUnadvise Method , page 525)
Implementation of the IDataObject.DUnAdvise method.

EnumDAdvise(see TVTDataObject.EnumDAdvise Method , page 525)
Implementation of the IDataObject.EnumDAdvise method.

EnumFormatEtc(see TVTDataObject.EnumFormatEtc Method , page 525)
Implementation of the IDataObject.EnumFormatEtc method.

EqualFormatEtc(see TVTDataObject.EqualFormatEtc Method , page 526)
Compares two TFormatEtc structures.

FindFormatEtc(see TVTDataObject.FindFormatEtc Method , page 526)
Searchs the given array for a the given format.

FindInternalStgMedium(see TVTDataObject.FindInternalStgMedium Method , page 526)
Returns a storage medium for a given clipboard format.

GetCanonicalFormatEtc(see TVTDataObject.GetCanonicalFormatEtc Method , page 526)
Implementation of the IDataObject.GetCanonicalFormatEtc method.

GetData(see TVTDataObject.GetData Method , page 527)
Implementation of the IDataObject.GetData method.

GetDataHere(see TVTDataObject.GetDataHere Method , page 527)
Implementation of the IDataObject.GetDataHere method.

HGlobalClone(see TVTDataObject.HGlobalClone Method , page 527)
Helper method for SetData(see TVTDataObject.SetData Method , page 529).

QueryGetData(see TVTDataObject.QueryGetData Method , page 528)
Implementation of the IDataObject.QueryGetData method.

RenderInternalOLEData(see TVTDataObject.RenderInternalOLEData Method , page 528)
Helper method to return data previously stored by SetData(see TVTDataObject.SetData Method , page 529).

SetData(see TVTDataObject.SetData Method , page 529)
Implementation of the IDataObject.SetData method.

StgMediumIncRef(see TVTDataObject.StgMediumIncRef Method , page 529)
Central managing method to copy OLE data.

Properties

ForClipboard(see TVTDataObject.ForClipboard Property, page 523)
Not documented.

FormatEtcArray(see TVTDataObject.FormatEtcArray Property, page 523)
Not documented.

InternalStgMediumArray(see TVTDataObject.InternalStgMediumArray Property, page 523)
Not documented.

Owner(see TVTDataObject.Owner Property, page 523)
Not documented.

Legend

protected

Property

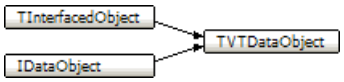
read only

Method

public

virtual

Class Hierarchy



File

VirtualTrees

10.1.22.1 TVTDataObject.ForClipboard Property

Not documented.

Pascal

```
property ForClipboard: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDataObject Class(see page 521)

10.1.22.2 TVTDataObject.FormatEtcArray Property

Not documented.

Pascal

```
property FormatEtcArray: TFormatEtcArray;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDataObject Class(see page 521)

10.1.22.3 TVTDataObject.InternalStgMediumArray Property

Not documented.

Pascal

```
property InternalStgMediumArray: TInternalStgMediumArray;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDataObject Class(see page 521)

10.1.22.4 TVTDataObject.Owner Property

Not documented.

Pascal

```
property Owner: TBaseVirtualTree;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDataObject Class(see page 521)

10.1.22.5 TVTDataObject.CanonicalIUnknown Method

Helper method for setting data in the IDataObject.

Pascal

```
function CanonicalIUnknown(TestUnknown: IUnknown): IUnknown;
```

Description

In SetData(see TVTDataObject.SetData Method , page 529) the class can get a circular reference if the client calls GetData(see TVTDataObject.GetData Method , page 527) then calls SetData(see TVTDataObject.SetData Method , page 529) with the same StgMedium. Because the unkForRelease for the IDataObject can be marshalled it is necessary to get pointers that can be correctly compared. CanonicalIUknown uses COM object identity for this task. An explicit call to the IUnknown::QueryInterface method, requesting the IUnknown interface, will always return the same pointer. See the IDragSourceHelper(see IDragSourceHelper Interface, page 701) article by Raymond Chen at MSDN.

Class

TVTDataObject Class(see page 521)

10.1.22.6 TVTDataObject.Create Constructor

Constructor of the class.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree; ForClipboard: Boolean); virtual;
```

Description

Create is used only for initialization.

Class

TVTDataObject Class(see page 521)

10.1.22.7 TVTDataObject.DAdvise Method

Implementation of the IDataObject.DAdvise method.

Pascal

```
function DAdvise(const FormatEtc: TFormatEtc; advf: Integer; const advSink: IAdviseSink; out
dwConnection: Integer): HResult; virtual; stdcall;
```

Description

Advise sinks are used to have an opportunity for clients to get notified if something changes in the data object. TVTDataObject(see TVTDataObject Class, page 521) uses the data advise holder APIs to provide the advise sink service.

Class

TVTDataObject Class(see page 521)

10.1.22.8 TVTDataObject.Destroy Destructor

Destructor of the class.

Pascal

```
destructor Destroy; override;
```

Description

Cleans up the object.

Class

TVTDataObject Class(see page 521)

10.1.22.9 TVTDataObject.DUnadvise Method

Implementation of the IDataObject.DUnAdvise method.

Pascal

```
function DUnadvise(dwConnection: Integer): HResult; virtual; stdcall;
```

Description

DUnadvice reverses the call to DAdvise(see TVTDataObject.DAdvise Method , page 524).

Class

TVTDataObject Class(see page 521)

10.1.22.10 TVTDataObject.EnumDAdvise Method

Implementation of the IDataObject.EnumDAdvise method.

Pascal

```
function EnumDAdvise(out enumAdvise: IEnumStatData): HResult; virtual; stdcall;
```

Description

EnumDAdvice does nothing but forwards the call to the internal advise holder class, which the responds accordingly. That's why we use data advise holders after all.

Class

TVTDataObject Class(see page 521)

10.1.22.11 TVTDataObject.EnumFormatEtc Method

Implementation of the IDataObject.EnumFormatEtc method.

Pascal

```
function EnumFormatEtc(Direction: Integer; out EnumFormatEtc: IEnumFormatEtc): HResult; virtual;
stdcall;
```

Description

This method creates a FormatEtc enumerator class which is used to enumerate all data formats supported by the owner tree.

Class

TVTDataObject Class(see page 521)

10.1.22.12 TVTDataObject.EqualFormatEtc Method

Compares two TFormatEtc structures.

Pascal

```
function EqualFormatEtc(FormatEtc1: TFormatEtc; FormatEtc2: TFormatEtc): Boolean;
```

Description

Returns true if both records are considered the same. That means if they have at least one common storage format and all other entries have the same values.

Class

TVTDataObject Class(see page 521)

10.1.22.13 TVTDataObject.FindFormatEtc Method

Searchs the given array for a the given format.

Pascal

```
function FindFormatEtc(TestFormatEtc: TFormatEtc; const FormatEtcArray: TFormatEtcArray): integer;
```

Description

Returns true if the given format is part of the array.

Class

TVTDataObject Class(see page 521)

10.1.22.14 TVTDataObject.FindInternalStgMedium Method

Returns a storage medium for a given clipboard format.

Pascal

```
function FindInternalStgMedium(Format: TClipFormat): PStgMedium;
```

Description

The class keeps an internal list of clipboard format/storage medium relations. For some operations data is set in certain formats which is later retrieve by locating it using this method.

Class

TVTDataObject Class(see page 521)

10.1.22.15 TVTDataObject.GetCanonicalFormatEtc Method

Implementation of the IDataObject.GetCanonicalFormatEtc method.

Pascal

```
function GetCanonicalFormatEtc(const FormatEtc: TFormatEtc; out FormatEtcOut: TFormatEtc): HResult;
virtual; stdcall;
```

Description

The implementation of this method simply consists of a result value telling the caller to use the EnumFormatEtc(see TVTDataObject.EnumFormatEtc Method , page 525) method.

Class

TVTDataObject Class(see page 521)

10.1.22.16 TVTDataObject.GetData Method

Implementation of the IDataObject.GetData method.

Pascal

```
function GetData(const FormatEtcIn: TFormatEtc; out Medium: TStgMedium): HResult; virtual; stdcall;
```

Description

Whenever drag'n drop or clipboard data actually needs to be rendered then this method is called by the OLE subsystem. The class automatically returns the CF_VTREFERENCE(see CF_VTREFERENCE Variable, page 659) format and any data previously set by the SetData(see TVTDataObject.SetData Method , page 529) method (e.g. by the Shell). For any other format the owner tree is asked to render the OLE data.

See Also

RenderOLEData

Class

TVTDataObject Class(see page 521)

10.1.22.17 TVTDataObject.GetDataHere Method

Implementation of the IDataObject.GetDataHere method.

Pascal

```
function GetDataHere(const FormatEtc: TFormatEtc; out Medium: TStgMedium): HResult; virtual; stdcall;
```

Description

GetDataHere is an alternative data retrival method to GetData(see TVTDataObject.GetData Method , page 527), but the caller provides the storage place where to store the actual data. Since Virtual Treeview has a very limited spectrum of what it can use this method is not fully implmented.

Class

TVTDataObject Class(see page 521)

10.1.22.18 TVTDataObject.HGlobalClone Method

Helper method for SetData(see TVTDataObject.SetData Method , page 529).

Pascal

```
function HGlobalClone(HGlobal: THandle): THandle;
```

Description

This method copies a HGlobal memory block.

Class

TVTDataObject Class(see page 521)

10.1.22.19 TVTDataObject.QueryGetData Method

Implementation of the IDataObject.QueryGetData method.

Pascal

```
function QueryGetData(const FormatEtc: TFormatEtc): HResult; virtual; stdcall;
```

Description

This method is called by OLE subsystem to determine which data formats are offered by the owner tree. It uses the internal clipboard format list to get a list of available and allowed formats. Currently following formats are supported:

TBaseVirtualTree(see TBaseVirtualTree Class, page 88)

- Virtual Treeview reference and process identifier
- native serialized tree data

TCustomVirtualStringTree(see TCustomVirtualStringTree Class, page 274)

- generic Unicode text
- generic ANSI text
- HTML formatted text (UTF-8 format)
- RTF text (UTF-16 format)
- CSV (comma separated values) but with customizable separators

Class

TVTDataObject Class(see page 521)

10.1.22.20 TVTDataObject.RenderInternalOLEData Method

Helper method to return data previously stored by SetData(see TVTDataObject.SetData Method , page 529).

Pascal

```
function RenderInternalOLEData(const FormatEtcIn: TFormatEtc; var Medium: TStgMedium; var OLEResult:
HResult): Boolean;
```

Description

For some operations (e.g. shell transfers with IDropTargetHelper(see IDropTargetHelper Interface, page 702) interface) data is stored in the class. RenderInternalOLEData returns this data when queried later.

Class

TVTDataObject Class(see page 521)

10.1.22.21 TVTDataObject.SetData Method

Implementation of the IDataObject.SetData method.

Pascal

```
function SetData(const FormatEtc: TFormatEtc; var Medium: TStgMedium; DoRelease: BOOL): HResult;
virtual; stdcall;
```

Description

This method is used to add or replace data in the data object.

Class

TVTDataObject Class(see page 521)

10.1.22.22 TVTDataObject.StgMediumIncRef Method

Central managing method to copy OLE data.

Pascal

```
function StgMediumIncRef(const InStgMedium: TStgMedium; var OutStgMedium: TStgMedium; CopyInMedium:
Boolean; DataObject: IDataObject): HRESULT;
```

Description

This method is called when data must be copied from or to the data object. For each supported storage medium a different (and appropriate) action is taken.

Class

TVTDataObject Class(see page 521)

10.1.23 TVTDragImage Class

Not documented.

Pascal

```
TVTDragImage = class;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes(see page 86)

Methods

Create(see TVTDragImage.Create Constructor , page 532)
Not documented.

Destroy(see TVTDragImage.Destroy Destructor , page 533)
Not documented.

DragTo(see TVTDragImage.DragTo Method , page 533)
Moves the drag image to a new position, which is determined from the passed point P and the previous mouse position.

EndDrag(see TVTDragImage.EndDrag Method , page 533)
Not documented.

GetDragImageRect(see TVTDragImage.GetDragImageRect Method , page 533)
Returns the current size and position of the drag image (screen coordinates).

HideDragImage(see TVTDragImage.HideDragImage Method , page 534)
Not documented.

InternalShowDragImage(see TVTDragImage.InternalShowDragImage Method , page 534)
Frequently called helper routine to actually do the blend and put it onto

MakeAlphaChannel(see TVTDragImage.MakeAlphaChannel Method , page 534)
Not documented.

PrepareDrag(see TVTDragImage.PrepareDrag Method , page 534)
Creates all necessary structures to do alpha blended dragging using the given image.

RecaptureBackground(see TVTDragImage.RecaptureBackground Method , page 535)
Notification by the drop target tree to update the background image because something in the tree has changed.

ShowDragImage(see TVTDragImage.ShowDragImage Method , page 535)
Shows the drag image after it has been hidden by HideDragImage(see TVTDragImage.HideDragImage Method , page 534).

WillMove(see TVTDragImage.WillMove Method , page 535)
Add a summary here...

Properties

ColorKey(see TVTDragImage.ColorKey Property, page 531)
Not documented.

Fade(see TVTDragImage.Fade Property, page 531)
Not documented.

MoveRestriction(see TVTDragImage.MoveRestriction Property, page 531)
Not documented.

PostBlendBias(see TVTDragImage.PostBlendBias Property, page 531)
Not documented.

PreBlendBias(see TVTDragImage.PreBlendBias Property, page 532)
Not documented.

Transparency(see TVTDragImage.Transparency Property, page 532)
Not documented.

Visible(see TVTDragImage.Visible Property, page 532)
Not documented.

Legend

public

Property

read only

Method

virtual

protected

Class Hierarchy

TVTDragImage

File

VirtualTrees

10.1.23.1 TVTDragImage.ColorKey Property

Not documented.

Pascal

```
property ColorKey: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.2 TVTDragImage.Fade Property

Not documented.

Pascal

```
property Fade: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.3 TVTDragImage.MoveRestriction Property

Not documented.

Pascal

```
property MoveRestriction: TVTDragMoveRestriction;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.4 TVTDragImage.PostBlendBias Property

Not documented.

Pascal

```
property PostBlendBias: TVTBias;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.5 TVTDragImage.PreBlendBias Property

Not documented.

Pascal

```
property PreBlendBias: TVTBias;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.6 TVTDragImage.Transparency Property

Not documented.

Pascal

```
property Transparency: TVTTransparency;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.7 TVTDragImage.Visible Property

Not documented.

Pascal

```
property Visible: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.8 TVTDragImage.Create Constructor

Not documented.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.9 TVTDragImage.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.10 TVTDragImage.DragTo Method

Moves the drag image to a new position, which is determined from the passed point P and the previous mouse position.

Pascal

```
function DragTo(P: TPoint; ForceRepaint: Boolean): Boolean;
```

Description

ForceRepaint is true if something on the screen changed and the back image must be refreshed.

Class

TVTDragImage Class(see page 529)

10.1.23.11 TVTDragImage.EndDrag Method

Not documented.

Pascal

```
procedure EndDrag;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.12 TVTDragImage.GetDragImageRect Method

Returns the current size and position of the drag image (screen coordinates).

Pascal

```
function GetDragImageRect: TRect;
```

Description

Returns the current size and position of the drag image (screen coordinates).

Class

TVTDragImage Class(see page 529)

10.1.23.13 TVTDragImage.HideDragImage Method

Not documented.

Pascal

```
procedure HideDragImage;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.14 TVTDragImage.InternalShowDragImage Method

Frequently called helper routine to actually do the blend and put it onto

Pascal

```
procedure InternalShowDragImage(ScreenDC: HDC);
```

Description

Frequently called helper routine to actually do the blend and put it onto the screen. Only used if the system does not support drag images.

Class

TVTDragImage Class(see page 529)

10.1.23.15 TVTDragImage.MakeAlphaChannel Method

Not documented.

Pascal

```
procedure MakeAlphaChannel(Source: TBitmap; Target: TBitmap);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class(see page 529)

10.1.23.16 TVTDragImage.PrepareDrag Method

Creates all necessary structures to do alpha blended dragging using the given image.

Pascal

```
procedure PrepareDrag(DragImage: TBitmap; ImagePosition: TPoint; HotSpot: TPoint; const DataObject:
IDataObject);
```

Description

ImagePostion and Hotspot are given in screen coordinates. The first determines where to place the drag image while the second is the initial mouse position. This method also determines whether the system supports drag images natively. If so then only minimal structures are created.

Class

TVTDragImage Class(see page 529)

10.1.23.17 TVTDragImage.RecaptureBackground Method

Notification by the drop target tree to update the background image because something in the tree has changed.

Pascal

```
procedure RecaptureBackground(Tree: TBaseVirtualTree; R: TRect; VisibleRegion: HRGN; CaptureNCArea:
Boolean; ReshowDragImage: Boolean);
```

Notes

The passed rectangle is given in client coordinates of the current drop target tree (given in Tree). The caller does not check if the given rectangle is actually within the drag image. Hence this method must do all the checks. This method does nothing if the system manages the drag image.

Class

TVTDragImage Class(see page 529)

10.1.23.18 TVTDragImage.ShowDragImage Method

Shows the drag image after it has been hidden by HideDragImage(see TVTDragImage.HideDragImage Method , page 534).

Pascal

```
procedure ShowDragImage;
```

Description

Also this method does nothing if the system manages the drag image.

Class

TVTDragImage Class(see page 529)

10.1.23.19 TVTDragImage.WillMove Method

Add a summary here...

Pascal

```
function WillMove(P: TPoint): Boolean;
```

Description

This method determines whether the drag image would "physically" move when DragTo(see TVTDragImage.DragTo Method , page 533) would be called with the same target point. Always returns false if the system drag image support is available.

Class

TVTDragImage Class(see page 529)

10.1.24 TVTDragManager Class

Not documented.

Pascal

```
TVTDragManager = class(TInterfacedObject, IVTDragManager, IDropSource, IDropTarget);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes(see page 86)

Methods

Create(see TVTDragManager.Create Constructor , page 537)
Not documented.

Destroy(see TVTDragManager.Destroy Destructor , page 537)
Not documented.

DragEnter(see TVTDragManager.DragEnter Method , page 538)
Not documented.

DragLeave(see TVTDragManager.DragLeave Method , page 538)
Not documented.

DragOver(see TVTDragManager.DragOver Method , page 538)
Not documented.

Drop(see TVTDragManager.Drop Method , page 538)
Not documented.

ForceDragLeave(see TVTDragManager.ForceDragLeave Method , page 539)
This method calls the drop(see TVTDragManager.Drop Method , page 538) target helper's DragLeave(see TVTDragManager.DragLeave Method , page 538) method to ensure it removes the drag image from screen.

GiveFeedback(see TVTDragManager.GiveFeedback Method , page 539)
Not documented.

QueryContinueDrag(see TVTDragManager.QueryContinueDrag Method , page 539)
Not documented.

IVTDragManager Interface

ForceDragLeave(see IVTDragManager.ForceDragLeave Method , page 706)
Not documented.

GetDataObject(see IVTDragManager.GetDataObject Method , page 706)
Not documented.

GetDragSource(see IVTDragManager.GetDragSource Method , page 707)
Not documented.

GetDropTargetHelperSupported(see IVTDragManager.GetDropTargetHelperSupported Method , page 707)
Not documented.

GetIsDropTarget(see IVTDragManager.GetIsDropTarget Method , page 707)
Not documented.

Properties

IVTDragManager Interface

DataObject(see IVTDragManager.DataObject Property, page 705)
Not documented.

DragSource(see IVTDragManager.DragSource Property, page 705)

Not documented.

DropTargetHelperSupported(see IVTDragManager.DropTargetHelperSupported Property, page 706)

Not documented.

IsDropTarget(see IVTDragManager.IsDropTarget Property, page 706)

Not documented.

Legend

public

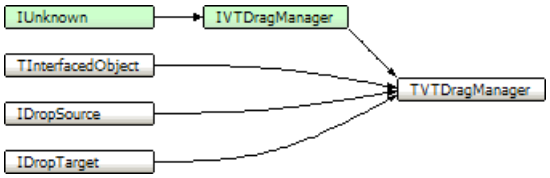
Method

virtual

Property

read only

Class Hierarchy



File

VirtualTrees

10.1.24.1 TVTDragManager.Create Constructor

Not documented.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class(see page 536)

10.1.24.2 TVTDragManager.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class(see page 536)

10.1.24.3 TVTDragManager.DragEnter Method

Not documented.

Pascal

```
function DragEnter(const DataObject: IDataObject; KeyState: Integer; Pt: TPoint; var Effect: Integer):
HResult; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class(see page 536)

10.1.24.4 TVTDragManager.DragLeave Method

Not documented.

Pascal

```
function DragLeave: HResult; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class(see page 536)

10.1.24.5 TVTDragManager.DragOver Method

Not documented.

Pascal

```
function DragOver(KeyState: Integer; Pt: TPoint; var Effect: Integer): HResult; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class(see page 536)

10.1.24.6 TVTDragManager.Drop Method

Not documented.

Pascal

```
function Drop(const DataObject: IDataObject; KeyState: Integer; Pt: TPoint; var Effect: Integer):
HResult; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class(see page 536)

10.1.24.7 TVTDragManager.ForceDragLeave Method

This method calls the drop(see TVTDragManager.Drop Method , page 538) target helper's DragLeave(see TVTDragManager.DragLeave Method , page 538) method to ensure it removes the drag image from screen.

Pascal

```
procedure ForceDragLeave; stdcall;
```

Description

This method calls the drop(see TVTDragManager.Drop Method , page 538) target helper's DragLeave(see TVTDragManager.DragLeave Method , page 538) method to ensure it removes the drag image from screen.

Class

TVTDragManager Class(see page 536)

10.1.24.8 TVTDragManager.GiveFeedback Method

Not documented.

Pascal

```
function GiveFeedback(Effect: Integer): HResult; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class(see page 536)

10.1.24.9 TVTDragManager.QueryContinueDrag Method

Not documented.

Pascal

```
function QueryContinueDrag(EscapePressed: BOOL; KeyState: Integer): HResult; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class(see page 536)

10.1.25 TVTEdit Class

Not documented.

Pascal

```
TVTEdit = class(TCustomEdit);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes(see page 86)

Methods

AutoAdjustSize(see TVTEdit.AutoAdjustSize Method , page 543)
Not documented.

Create(see TVTEdit.Create Constructor , page 543)
Not documented.

CreateParams(see TVTEdit.CreateParams Method , page 543)
Not documented.

Release(see TVTEdit.Release Method , page 543)
Not documented.

Properties

AutoSelect(see TVTEdit.AutoSelect Property, page 541)
Not documented.

AutoSize(see TVTEdit.AutoSize Property, page 541)
Not documented.

BorderStyle(see TVTEdit.BorderStyle Property, page 541)
Not documented.

CharCase(see TVTEdit.CharCase Property, page 541)
Not documented.

HideSelection(see TVTEdit.HideSelection Property, page 542)
Not documented.

MaxLength(see TVTEdit.MaxLength Property, page 542)
Not documented.

OEMConvert(see TVTEdit.OEMConvert Property, page 542)
Not documented.

PasswordChar(see TVTEdit.PasswordChar Property, page 542)
Not documented.

Legend

public

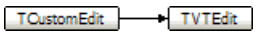
Property

protected

Method

virtual

Class Hierarchy



File

VirtualTrees

10.1.25.1 TVTEdit.AutoSelect Property

Not documented.

Pascal

```
property AutoSelect;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.2 TVTEdit.AutoSize Property

Not documented.

Pascal

```
property AutoSize;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.3 TVTEdit.BorderStyle Property

Not documented.

Pascal

```
property BorderStyle;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.4 TVTEdit.CharCase Property

Not documented.

Pascal

```
property CharCase;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.5 TVTEdit.HideSelection Property

Not documented.

Pascal

```
property HideSelection;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.6 TVTEdit.MaxLength Property

Not documented.

Pascal

```
property MaxLength;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.7 TVTEdit.OEMConvert Property

Not documented.

Pascal

```
property OEMConvert;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.8 TVTEdit.PasswordChar Property

Not documented.

Pascal

```
property PasswordChar;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.9 TVTEdit.AutoAdjustSize Method

Not documented.

Pascal

```
procedure AutoAdjustSize;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.10 TVTEdit.Create Constructor

Not documented.

Pascal

```
constructor Create(Link: TStringEditLink); reintroduce;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.11 TVTEdit.CreateParams Method

Not documented.

Pascal

```
procedure CreateParams(var Params: TCreateParams); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.25.12 TVTEdit.Release Method

Not documented.

Pascal

```
procedure Release; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class(see page 539)

10.1.26 TVTHeader Class

Not documented.

Pascal

```
TVTHeader = class(TPersistent);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes(see page 86)

Methods

Assign(see TVTHeader.Assign Method , page 550)
Not documented.

AutoFitColumns(see TVTHeader.AutoFitColumns Method , page 550)
Not documented.

CanWriteColumns(see TVTHeader.CanWriteColumns Method , page 551)
Not documented.

ChangeScale(see TVTHeader.ChangeScale Method , page 551)
Not documented.

Create(see TVTHeader.Create Constructor , page 551)
Not documented.

Destroy(see TVTHeader.Destroy Destructor , page 551)
Not documented.

DetermineSplitterIndex(see TVTHeader.DetermineSplitterIndex Method , page 552)
Tries to find the index of that column whose right border corresponds to P.

DragTo(see TVTHeader.DragTo Method , page 552)
Moves the drag image to a new position, which is determined from the passed point P and the previous mouse position.

GetColumnsClass(see TVTHeader.GetColumnsClass Method , page 552)
Returns the class to be used for the actual column implementation.

GetOwner(see TVTHeader.GetOwner Method , page 553)
Not documented.

GetShiftState(see TVTHeader.GetShiftState Method , page 553)
Not documented.

HandleHeaderMouseMove(see TVTHeader.HandleHeaderMouseMove Method , page 553)
Not documented.

HandleMessage(see TVTHeader.HandleMessage Method , page 553)
General message handler for the header.

ImageListChange(see TVTHeader.ImageListChange Method , page 554)
Not documented.

InHeader(see TVTHeader.InHeader Method , page 554)
Determines whether the given point (client coordinates!) is within the header rectangle (non-client coordinates).

Invalidate(see TVTHeader.Invalidate Method , page 554)
Invalidates the entire header or parts of it so they are repainted.

LoadFromStream(see TVTHeader.LoadFromStream Method , page 554)
Restores the state of the header from the given stream.

PrepareDrag(see TVTHeader.PrepareDrag Method , page 555)
Initializes dragging of the header, P is the current mouse postion and Start the initial mouse position.

ReadColumns(see TVTHeader.ReadColumns Method , page 555)
Not documented.

RecalculateHeader(see TVTHeader.RecalculateHeader Method , page 555)
Initiate a recalculation of the non-client area of the owner tree.

RestoreColumns(see TVTHeader.RestoreColumns Method , page 555)
Restores all columns to their width which they had before they have been auto fitted.

SaveToStream(see TVTHeader.SaveToStream Method , page 556)
Saves the complete state of the header into the provided stream.

UpdateMainColumn(see TVTHeader.UpdateMainColumn Method , page 556)
Called once the load process of the owner tree is done.

UpdateSpringColumns(see TVTHeader.UpdateSpringColumns Method , page 556)
Not documented.

WriteColumns(see TVTHeader.WriteColumns Method , page 556)
Not documented.

Properties

AutoSizeIndex(see TVTHeader.AutoSizeIndex Property, page 546)
Not documented.

Background(see TVTHeader.Background Property, page 546)
Not documented.

Columns(see TVTHeader.Columns Property, page 546)
Not documented.

DragImage(see TVTHeader.DragImage Property, page 547)
Not documented.

Font(see TVTHeader.Font Property, page 547)
Not documented.

Height(see TVTHeader.Height Property, page 547)
Not documented.

Images(see TVTHeader.Images Property, page 547)
Not documented.

MainColumn(see TVTHeader.MainColumn Property, page 548)
Not documented.

Options(see TVTHeader.Options Property, page 548)
Not documented.

ParentFont(see TVTHeader.ParentFont Property, page 548)
Not documented.

PopupMenu(see TVTHeader.PopupMenu Property, page 548)
Not documented.

SortColumn(see TVTHeader.SortColumn Property, page 549)
Not documented.

SortDirection(see TVTHeader.SortDirection Property, page 549)
Not documented.

States(see TVTHeader.States Property, page 549)
Not documented.

Style(see TVTHeader.Style Property, page 549)
Not documented.

Treeview(see TVTHeader.Treeview Property, page 550)
Not documented.

UseColumns(see TVTHeader.UseColumns Property, page 550)
Not documented.

Legend


published


Property


public

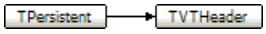

read only


Method


virtual


protected

Class Hierarchy



File

VirtualTrees

10.1.26.1 TVTHeader.AutoSizeIndex Property

Not documented.

Pascal

```
property AutoSizeIndex: TColumnIndex;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.2 TVTHeader.Background Property

Not documented.

Pascal

```
property Background: TColor;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.3 TVTHeader.Columns Property

Not documented.

Pascal

```
property Columns: TVirtualTreeColumns;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.4 TVTHeader.DragImage Property

Not documented.

Pascal

```
property DragImage: TVTDragImage;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.5 TVTHeader.Font Property

Not documented.

Pascal

```
property Font: TFont;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.6 TVTHeader.Height Property

Not documented.

Pascal

```
property Height: Cardinal;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.7 TVTHeader.Images Property

Not documented.

Pascal

```
property Images: TCustomImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.8 TVTHeader.MainColumn Property

Not documented.

Pascal

```
property MainColumn: TColumnIndex;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.9 TVTHeader.Options Property

Not documented.

Pascal

```
property Options: TVTHeaderOptions;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.10 TVTHeader.ParentFont Property

Not documented.

Pascal

```
property ParentFont: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.11 TVTHeader.PopupMenu Property

Not documented.

Pascal

```
property PopupMenu: TPopupMenu;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.12 TVTHeader.SortColumn Property

Not documented.

Pascal

```
property SortColumn: TColumnIndex;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.13 TVTHeader.SortDirection Property

Not documented.

Pascal

```
property SortDirection: TSortDirection;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.14 TVTHeader.States Property

Not documented.

Pascal

```
property States: THeaderStates;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.15 TVTHeader.Style Property

Not documented.

Pascal

```
property Style: TVTHeaderStyle;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.16 TVTHeader.Treeview Property

Not documented.

Pascal

```
property Treeview: TBaseVirtualTree;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.17 TVTHeader.UseColumns Property

Not documented.

Pascal

```
property UseColumns: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.18 TVTHeader.Assign Method

Not documented.

Pascal

```
procedure Assign(Source: TPersistent); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.19 TVTHeader.AutoFitColumns Method

Not documented.

Pascal

```
procedure AutoFitColumns(Animated: Boolean = True);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.20 TVTHeader.CanWriteColumns Method

Not documented.

Pascal

```
function CanWriteColumns: Boolean; virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.21 TVTHeader.ChangeScale Method

Not documented.

Pascal

```
procedure ChangeScale(M: Integer; D: Integer); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.22 TVTHeader.Create Constructor

Not documented.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.23 TVTHeader.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.24 TVTHeader.DetermineSplitterIndex Method

Tries to find the index of that column whose right border corresponds to P.

Pascal

```
function DetermineSplitterIndex(P: TPoint): Boolean; virtual;
```

Description

Result is rrue if column border was hit (with -3..+5 pixels tolerance). For continuous resizing the current track index and the column's left border are set.

Notes

The hit test is checking from right to left to make enlarging of zero-sized columns possible.

Class

TVTHeader Class(see page 544)

10.1.26.25 TVTHeader.DragTo Method

Moves the drag image to a new position, which is determined from the passed point P and the previous mouse position.

Pascal

```
procedure DragTo(P: TPoint);
```

Description

Moves the drag image to a new position, which is determined from the passed point P and the previous mouse position.

Class

TVTHeader Class(see page 544)

10.1.26.26 TVTHeader.GetColumnsClass Method

Returns the class to be used for the actual column implementation.

Pascal

```
function GetColumnsClass: TVirtualTreeColumnsClass; virtual;
```

Description

Descentants may optionally override this and return their own class.

Class

TVTHeader Class(see page 544)

10.1.26.27 TVTHeader.GetOwner Method

Not documented.

Pascal

```
function GetOwner: TPersistent; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.28 TVTHeader.GetShiftState Method

Not documented.

Pascal

```
function GetShiftState: TShiftState;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.29 TVTHeader.HandleHeaderMouseMove Method

Not documented.

Pascal

```
function HandleHeaderMouseMove(var Message: TWMMouseMove): Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.30 TVTHeader.HandleMessage Method

General message handler for the header.

Pascal

```
function HandleMessage(var Message: TMessage): Boolean; virtual;
```

Description

The header gets here the opportunity to handle certain messages before they reach the tree. This is important because the tree needs to handle various non-client area messages for the header as well as some dragging/tracking events. By returning True the message will not be handled further, otherwise the message is then dispatched to the proper message handlers.

Class

TVTHeader Class(see page 544)

10.1.26.31 TVTHeader.ImageListChange Method

Not documented.

Pascal

```
procedure ImageListChange(Sender: TObject);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.32 TVTHeader.InHeader Method

Determines whether the given point (client coordinates!) is within the header rectangle (non-client coordinates).

Pascal

```
function InHeader(P: TPoint): Boolean; virtual;
```

Description

Determines whether the given point (client coordinates!) is within the header rectangle (non-client coordinates).

Class

TVTHeader Class(see page 544)

10.1.26.33 TVTHeader.Invalidate Method

Invalidates the entire header or parts of it so they are repainted.

Pascal

```
procedure Invalidate(Column: TVirtualTreeColumn; ExpandToRight: Boolean = False);
```

Description

Because the header is in the non-client area of the tree it needs some special handling in order to initiate its repainting. If ExpandToRight is true then not only the given column but everything to its right will be invalidated (useful for resizing). This makes only sense when a column is given.

Class

TVTHeader Class(see page 544)

10.1.26.34 TVTHeader.LoadFromStream Method

Restores the state of the header from the given stream.

Pascal

```
procedure LoadFromStream(const Stream: TStream); virtual;
```

Description

Restores the state of the header from the given stream.

Class

TVTHeader Class(see page 544)

10.1.26.35 TVTHeader.PrepareDrag Method

Initializes dragging of the header, P is the current mouse postion and Start the initial mouse position.

Pascal

```
procedure PrepareDrag(P: TPoint; Start: TPoint);
```

Description

Initializes dragging of the header, P is the current mouse postion and Start the initial mouse position.

Class

TVTHeader Class(see page 544)

10.1.26.36 TVTHeader.ReadColumns Method

Not documented.

Pascal

```
procedure ReadColumns(Reader: TReader);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.37 TVTHeader.RecalculateHeader Method

Initiate a recalculation of the non-client area of the owner tree.

Pascal

```
procedure RecalculateHeader; virtual;
```

Description

Initiate a recalculation of the non-client area of the owner tree.

Class

TVTHeader Class(see page 544)

10.1.26.38 TVTHeader.RestoreColumns Method

Restores all columns to their width which they had before they have been auto fitted.

Pascal

```
procedure RestoreColumns;
```

Description

Restores all columns to their width which they had before they have been auto fitted.

Class

TVTHeader Class(see page 544)

10.1.26.39 TVTHeader.SaveToStream Method

Saves the complete state of the header into the provided stream.

Pascal

```
procedure SaveToStream(const Stream: TStream); virtual;
```

Description

Saves the complete state of the header into the provided stream.

Class

TVTHeader Class(see page 544)

10.1.26.40 TVTHeader.UpdateMainColumn Method

Called once the load process of the owner tree is done.

Pascal

```
procedure UpdateMainColumn;
```

Description

Called once the load process of the owner tree is done.

Class

TVTHeader Class(see page 544)

10.1.26.41 TVTHeader.UpdateSpringColumns Method

Not documented.

Pascal

```
procedure UpdateSpringColumns;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.26.42 TVTHeader.WriteColumns Method

Not documented.

Pascal

```
procedure WriteColumns(Writer: TWriter);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class(see page 544)

10.1.27 TVTHeaderPopupMenu Class

Not documented.

Pascal

```
TVTHeaderPopupMenu = class(TPopupMenu);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Events

OnAddHeaderPopupItem(see TVTHeaderPopupMenu.OnAddHeaderPopupItem Event, page 558)
Not documented.

OnColumnChange(see TVTHeaderPopupMenu.OnColumnChange Event, page 558)
Not documented.

Group

Classes(see page 86)

Methods

DoAddHeaderPopupItem(see TVTHeaderPopupMenu.DoAddHeaderPopupItem Method , page 558)
Not documented.

DoColumnChange(see TVTHeaderPopupMenu.DoColumnChange Method , page 559)
Not documented.

OnMenuItemClick(see TVTHeaderPopupMenu.OnMenuItemClick Method , page 559)
Not documented.

Popup(see TVTHeaderPopupMenu.Popup Method , page 559)
Not documented.

Properties

Options(see TVTHeaderPopupMenu.Options Property, page 558)
Not documented.

Legend


published


Event


Property


protected


Method


virtual

public

Class Hierarchy

File

VTHeaderPopup

10.1.27.1 TVTHeaderPopupMenu.OnAddHeaderPopupItem Event

Not documented.

Pascal

```
property OnAddHeaderPopupItem: TAddHeaderPopupItemEvent;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeaderPopupMenu Class(see page 557)

10.1.27.2 TVTHeaderPopupMenu.OnColumnChange Event

Not documented.

Pascal

```
property OnColumnChange: TColumnChangeEvent;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeaderPopupMenu Class(see page 557)

10.1.27.3 TVTHeaderPopupMenu.Options Property

Not documented.

Pascal

```
property Options: TVTHeaderPopupOptions;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeaderPopupMenu Class(see page 557)

10.1.27.4 TVTHeaderPopupMenu.DoAddHeaderPopupItem Method

Not documented.

Pascal

```
procedure DoAddHeaderPopupItem(const Column: TColumnIndex; out Cmd: TAddPopupItemType); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeaderPopupMenu Class(see page 557)

10.1.27.5 TVTHeaderPopupMenu.DoColumnChange Method

Not documented.

Pascal

```
procedure DoColumnChange(Column: TColumnIndex; Visible: Boolean); virtual;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeaderPopupMenu Class(see page 557)

10.1.27.6 TVTHeaderPopupMenu.OnMenuItemClick Method

Not documented.

Pascal

```
procedure OnMenuItemClick(Sender: TObject);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeaderPopupMenu Class(see page 557)

10.1.27.7 TVTHeaderPopupMenu.Popup Method

Not documented.

Pascal

```
procedure Popup(x: Integer; y: Integer); override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeaderPopupMenu Class(see page 557)

10.1.28 TWideBufferedString Class

Not documented.

Pascal

```
TWideBufferedString = class;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes(see page 86)

Methods

Add(see TWideBufferedString.Add Method , page 561)
Not documented.

AddNewLine(see TWideBufferedString.AddNewLine Method , page 561)
Not documented.

Destroy(see TWideBufferedString.Destroy Destructor , page 561)
Not documented.

Properties

AsString(see TWideBufferedString.AsString Property, page 560)
Not documented.

Legend

public

Property

read only

Method

virtual

Class Hierarchy

TWideBufferedString

File

VirtualTrees

10.1.28.1 TWideBufferedString.AsString Property

Not documented.

Pascal

```
property AsString: WideString;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWideBufferedString Class(see page 560)

10.1.28.2 TWideBufferedString.Add Method

Not documented.

Pascal

```
procedure Add(const S: WideString);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWideBufferedString Class(see page 560)

10.1.28.3 TWideBufferedString.AddNewLine Method

Not documented.

Pascal

```
procedure AddNewLine;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWideBufferedString Class(see page 560)

10.1.28.4 TWideBufferedString.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWideBufferedString Class(see page 560)

10.1.29 TWorkerThread Class

Not documented.

Pascal

```
TWorkerThread = class(TThread);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes(see page 86)

Methods

AddTree(see TWorkerThread.AddTree Method , page 563)
Not documented.

ChangeTreeStates(see TWorkerThread.ChangeTreeStates Method , page 563)
Not documented.

Create(see TWorkerThread.Create Constructor , page 563)
Not documented.

Destroy(see TWorkerThread.Destroy Destructor , page 563)
Not documented.

Execute(see TWorkerThread.Execute Method , page 564)
Not documented.

RemoveTree(see TWorkerThread.RemoveTree Method , page 564)
Not documented.

Properties

CurrentTree(see TWorkerThread.CurrentTree Property, page 562)
Not documented.

Legend

public

Property

read only

Method

protected

virtual

Class Hierarchy

File

VirtualTrees

10.1.29.1 TWorkerThread.CurrentTree Property

Not documented.

Pascal

```
property CurrentTree: TBaseVirtualTree;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWorkerThread Class(see page 561)

10.1.29.2 TWorkerThread.AddTree Method

Not documented.

Pascal

```
procedure AddTree(Tree: TBaseVirtualTree);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWorkerThread Class(see page 561)

10.1.29.3 TWorkerThread.ChangeTreeStates Method

Not documented.

Pascal

```
procedure ChangeTreeStates(EnterStates: TChangeStates; LeaveStates: TChangeStates);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWorkerThread Class(see page 561)

10.1.29.4 TWorkerThread.Create Constructor

Not documented.

Pascal

```
constructor Create(CreateSuspended: Boolean);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWorkerThread Class(see page 561)

10.1.29.5 TWorkerThread.Destroy Destructor

Not documented.

Pascal

```
destructor Destroy; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWorkerThread Class(see page 561)

10.1.29.6 TWorkerThread.Execute Method

Not documented.

Pascal

```
procedure Execute; override;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWorkerThread Class(see page 561)

10.1.29.7 TWorkerThread.RemoveTree Method

Not documented.

Pascal

```
procedure RemoveTree(Tree: TBaseVirtualTree);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWorkerThread Class(see page 561)

10.1.30 TWriterHack Class

Not documented.

Pascal

```
TWriterHack = class(TFiler);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes(see page 86)

Class Hierarchy

File

VirtualTrees

10.2 Functions

These are all functions that are contained in this documentation.

Functions

AlphaBlend(see AlphaBlend Function, page 565)
General purpose procedure to blend one bitmap to another.

DrawTextW(see DrawTextW Function, page 566)
Paint support procedure.

EnumerateVTClipboardFormats(see EnumerateVTClipboardFormats Function, page 567)
Not documented.

EnumerateVTClipboardFormats(see EnumerateVTClipboardFormats Function, page 567)
Not documented.

GetVTClipboardFormatDescription(see GetVTClipboardFormatDescription Function, page 567)
Not documented.

PrtStretchDrawDIB(see PrtStretchDrawDIB Function, page 568)
Not documented.

RegisterVTClipboardFormat(see RegisterVTClipboardFormat Function, page 568)
Methods to register a certain clipboard format for a given tree class.

RegisterVTClipboardFormat(see RegisterVTClipboardFormat Function, page 568)
Methods to register a certain clipboard format for a given tree class.

ShortenString(see ShortenString Function, page 568)
General purpose routine to shorten a Unicode string to a given maximum size.

TreeFromNode(see TreeFromNode Function, page 569)
General purpose routine to get the tree to which a node belongs.

Group

Virtual Treeview(see page 71)

Legend

Function

10.2.1 AlphaBlend Function

General purpose procedure to blend one bitmap to another.

Pascal

```
procedure AlphaBlend(Source: HDC; Destination: HDC; R: TRect; Target: TPoint; Mode: TBlendMode;
ConstantAlpha: Integer; Bias: Integer);
```

Description

This is an optimized alpha blend procedure using MMX instructions to perform as quick as possible. For this procedure to work properly it is important that both source and target bitmap use the 32 bit color format (pf32Bit for TBitmap). R describes the source rectangle to work on, while Target is the place (upper left corner) in the target bitmap where to blend to. Note that source width + X offset must be less or equal to the target width. Similar for the height.

If Mode is bmConstantAlpha then the blend operation uses the given ConstantAlpha value for all pixels.

If Mode is bmPerPixelAlpha then each pixel is blended using its individual alpha value (the alpha value of the source).

If Mode is bmMasterAlpha then each pixel is blended using its individual alpha value multiplied by ConstantAlpha.

If Mode is bmConstantAlphaAndColor then each destination pixel is blended using ConstantAlpha but also a constant color which will be obtained from Bias. In this case no offset value is added, otherwise Bias is used as offset.

Blending of a color into target only (bmConstantAlphaAndColor) ignores Source (the DC) and Target (the position).

Notes

This procedure does not check whether MMX instructions are actually available! Call it only if MMX is really usable,

otherwise a process exception for unknown op codes is thrown.

Group

Functions(see page 565)

File

VirtualTrees

10.2.2 DrawTextW Function

Paint support procedure.

Pascal

```
procedure DrawTextW(DC: HDC; lpString: PWideChar; nCount: Integer; var lpRect: TRect; uFormat:
Cardinal; AdjustRight: Boolean);
```

Description

This procedure implements a subset of Window's DrawText API for Unicode which is not available for Windows 95, 98 and ME. For a description of the parameters see DrawText in the online help.

Supported flags are currently:

- DT_LEFT
- DT_TOP
- DT_CALCRECT
- DT_NOCLIP
- DT_RTLREADING
- DT_SINGLELINE
- DT_VCENTER

Differences to the DrawTextW Windows API:

The additional parameter AdjustRight determines whether to adjust the right border of the given rectangle to accomodate the largest line in the text. It has only a meaning if also DT_CALCRECT is specified.

Notes

When running on any NT windows version (Windows NT 4.0, Windows 2000., Windows XP and up) the native windows API is used

instead of this method as it also supports word wrapping properly.

Group

Functions(see page 565)

File

VirtualTrees

10.2.3 EnumerateVTClipboardFormats Function

Not documented.

Pascal

```
procedure EnumerateVTClipboardFormats(TreeClass: TVirtualTreeClass; var Formats: TFormatEtcArray);
overload;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Functions(see page 565)

File

VirtualTrees

10.2.4 EnumerateVTClipboardFormats Function

Not documented.

Pascal

```
procedure EnumerateVTClipboardFormats(TreeClass: TVirtualTreeClass; const List: TStrings); overload;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Functions(see page 565)

File

VirtualTrees

10.2.5 GetVTClipboardFormatDescription Function

Not documented.

Pascal

```
function GetVTClipboardFormatDescription(AFormat: Word): string;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Functions(see page 565)

File

VirtualTrees

10.2.6 PrtStretchDrawDIB Function

Not documented.

Pascal

```
procedure PrtStretchDrawDIB(Canvas: TCanvas; DestRect: TRect; ABitmap: TBitmap);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Functions(see page 565)

File

VirtualTrees

10.2.7 RegisterVTClipboardFormat Function

Methods to register a certain clipboard format for a given tree class.

Pascal

```
procedure RegisterVTClipboardFormat(AFormat: Word; TreeClass: TVirtualTreeClass; Priority: Cardinal);
overload;
function RegisterVTClipboardFormat(Description: string; TreeClass: TVirtualTreeClass; Priority:
Cardinal; tymed: Integer = TYMED_HGLOBAL; ptd: PDVTargetDevice = nil; dwAspect: Integer =
DVASPECT_CONTENT; lindex: Integer = -1): Word; overload;
```

Description

Registration with the clipboard is done here too and the assigned ID returned by the function. tymed may contain or'ed TYMED constants which allows to register several storage formats for one clipboard format.

Group

Functions(see page 565)

File

VirtualTrees

10.2.8 ShortenString Function

General purpose routine to shorten a Unicode string to a given maximum size.

Pascal

```
function ShortenString(DC: HDC; const S: WideString; Width: Integer; RTL: Boolean; EllipsisWidth:
Integer = 0): WideString;
```

Description

Adjusts the given string S so that it fits into the given width. DC holds the handle to a valid device context which can be used to determine the width of a string. Of couse, this device context should be set up with the proper values for the current font. EllipsisWidth gives the width (in logical units) of the three points to be added to the shorted string. If this value is 0 then it will be determined implicitely. For higher speed (and multiple entries to be shorted) specify this value explicitely. RTL determines if right-to-left reading is active, which is needed to put the ellipsisis on the correct side. The result is the left part of the string which fits into the given space plus the ellipsisis.

Notes

It is assumed that the string really needs shortage. Check this in advance.

Group

Functions(see page 565)

File

VirtualTrees

10.2.9 TreeFromNode Function

General purpose routine to get the tree to which a node belongs.

Pascal

```
function TreeFromNode(Node: PVirtualNode): TBaseVirtualTree;
```

Description

For obvious reasons it makes no sense to store the reference to a tree in each node record, but sometimes there might arise the need to know to which tree a node belongs. This is not often the case but is necessary e.g. for optimized moves in drag'n drop or cut'n paste operations.

Each node contains a reference to its parent to allow fast traversal. The hidden root node, however, does not need this reference because it does not have a node parent. Instead it contains the reference of the tree to which it belongs. To determine which node is the root node (when you don't know its tree) a special case of sibling reference is used. Since the root node does neither have a previous nor a next sibling the corresponding pointers are set to the root node, making the root so pointing to itself. This case will never happen in "normal" nodes, so it is a reliable way to detect the root node.

Group

Functions(see page 565)

File

VirtualTrees

10.3 Structs and Records

These are all structs and records that are contained in this documentation.

Enumerations

TVTTooltipLineBreakStyle(see TVTTooltipLineBreakStyle Enumeration, page 579)
Not documented.

Group

Virtual Treeview(see page 71)

Records

TBaseChunk(see TBaseChunk Record, page 571)
Not documented.

TBaseChunkBody(see TBaseChunkBody Record, page 571)
Not documented.

TCacheEntry(see TCacheEntry Record, page 571)
Not documented.

TChunkHeader(see TChunkHeader Record, page 572)
Not documented.

TClipboardFormatEntry(see TClipboardFormatEntry Record, page 572)
Not documented.

TClipboardFormatListEntry(see TClipboardFormatListEntry Record, page 573)
Not documented.

THeaderPaintInfo(see THeaderPaintInfo Record, page 573)
Not documented.

THitInfo(see THitInfo Record, page 574)
Not documented.

TInternalStgMedium(see TInternalStgMedium Record, page 574)
Not documented.

TRealWMNCPaint(see TRealWMNCPaint Record, page 574)
Not documented.

TSHDragImage(see TSHDragImage Record, page 575)
Not documented.

TToggleAnimationData(see TToggleAnimationData Record, page 575)
Not documented.

TVirtualNode(see TVirtualNode Record, page 576)
Not documented.

TVTHintData(see TVTHintData Record, page 577)
Not documented.

TVTImageInfo(see TVTImageInfo Record, page 577)
Not documented.

TVTPaintInfo(see TVTPaintInfo Record, page 578)
Not documented.

TVTReference(see TVTReference Record, page 578)
Not documented.

TWMPrint(see TWMPrint Record, page 579)
Not documented.

Legend

Struct

10.3.1 TBaseChunk Record

Not documented.

Pascal

```
TBaseChunk = packed record
  Header: TChunkHeader;
  Body: TBaseChunkBody;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.2 TBaseChunkBody Record

Not documented.

Pascal

```
TBaseChunkBody = packed record
  ChildCount: Cardinal;
  NodeHeight: Cardinal;
  States: TVirtualNodeStates;
  Align: Byte;
  CheckState: TCheckState;
  CheckType: TCheckType;
  Reserved: Cardinal;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.3 TCacheEntry Record

Not documented.

Pascal

```
TCacheEntry = record
  Node: PVirtualNode;
  AbsoluteTop: Cardinal;
```

```
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.4 TChunkHeader Record

Not documented.

Pascal

```
TChunkHeader = record
  ChunkSize: Integer;
  ChunkType: Integer;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.5 TClipboardFormatEntry Record

Not documented.

Pascal

```
TClipboardFormatEntry = record
  ID: Word;
  Description: string;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.6 TClipboardFormatListEntry Record

Not documented.

Pascal

```
TClipboardFormatListEntry = record
  Description: string;
  TreeClass: TVirtualTreeClass;
  Priority: Cardinal;
  FormatEtc: TFormatEtc;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
`Description: string;`	The string used to register the format with Windows.
`TreeClass: TVirtualTreeClass;`	The tree class which supports rendering this format.
`Priority: Cardinal;`	Number which determines the order of formats used in IDataObject.
`FormatEtc: TFormatEtc;`	The definition of the format in the IDataObject.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.7 THeaderPaintInfo Record

Not documented.

Pascal

```
THeaderPaintInfo = record
  TargetCanvas: TCanvas;
  Column: TVirtualTreeColumn;
  PaintRectangle: TRect;
  TextRectangle: TRect;
  IsDownIndex: Boolean;
  IsEnabled: Boolean;
  IsHoverIndex: Boolean;
  ShowHeaderGlyph: Boolean;
  ShowRightBorder: Boolean;
  ShowSortGlyph: Boolean;
  DropMark: TVTDropMarkMode;
  GlyphPos: TPoint;
  SortGlyphPos: TPoint;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.8 THitInfo Record

Not documented.

Pascal

```
THitInfo = record
  HitNode: PVirtualNode;
  HitPositions: THitPositions;
  HitColumn: TColumnIndex;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.9 TInternalStgMedium Record

Not documented.

Pascal

```
TInternalStgMedium = packed record
  Format: TClipFormat;
  Medium: TStgMedium;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.10 TRealWMNCPaint Record

Not documented.

Pascal

```
TRealWMNCPaint = packed record
  Msg: Cardinal;
  Rgn: HRGN;
```

```
  lParam: Integer;
  Result: Integer;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.11 TSHDragImage Record

Not documented.

Pascal

```
TSHDragImage = packed record
  sizeDragImage: TSize;
  ptOffset: TPoint;
  hbmpDragImage: HBITMAP;
  ColorRef: TColorRef;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.12 TToggleAnimationData Record

Not documented.

Pascal

```
TToggleAnimationData = record
  Expand: Boolean;
  Window: HWND;
  DC: HDC;
  Brush: HBRUSH;
  R: TRect;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
`Expand: Boolean;`	if true then expanding is in progress
`Window: HWND;`	copy of the tree's window handle

DC: HDC;	the DC of the window to erase unconvered parts
Brush: HBRUSH;	the brush to be used to erase uncovered parts
R: TRect;	the scroll rectangle

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.13 TVirtualNode Record

Not documented.

Pascal

```
TVirtualNode = packed record
  ChildCount: Cardinal;
  Index: Cardinal;
  NodeHeight: Word;
  States: TVirtualNodeStates;
  Align: Byte;
  CheckState: TCheckState;
  CheckType: TCheckType;
  Dummy: Byte;
  TotalCount: Cardinal;
  TotalHeight: Cardinal;
  FirstChild: PVirtualNode;
  LastChild: PVirtualNode;
  NextSibling: PVirtualNode;
  Parent: PVirtualNode;
  PrevSibling: PVirtualNode;
  Data: record;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
NodeHeight: Word;	height in pixels
States: TVirtualNodeStates;	states describing various properties of the node (expanded, initialized etc.)
Align: Byte;	line/button alignment
CheckState: TCheckState;	checked, pressed etc.)
CheckType: TCheckType;	indicates which check type shall be used for this node
Dummy: Byte;	dummy value to fill DWORD boundary
Data: record;	this is a placeholder, each node gets extra data determined by NodeDataSize

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.14 TVTHintData Record

Not documented.

Pascal

```
TVTHintData = record
  Tree: TBaseVirtualTree;
  Node: PVirtualNode;
  Column: TColumnIndex;
  HintRect: TRect;
  DefaultHint: WideString;
  HintText: WideString;
  BidiMode: TBidiMode;
  Alignment: TAlignment;
  LineBreakStyle: TVTToolTipLineBreakStyle;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
`HintRect: TRect;`	used for draw trees only, string trees get the size from the hint string
`DefaultHint: WideString;`	used only if there is no node specific hint string available or a header hint is about to appear
`HintText: WideString;`	set when size of the hint window is calculated

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.15 TVTImageInfo Record

Not documented.

Pascal

```
TVTImageInfo = record
  Index: Integer;
  XPos: Integer;
  YPos: Integer;
  Ghosted: Boolean;
  Images: TCustomImageList;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
`Index: Integer;`	index in the associated image list
`Ghosted: Boolean;`	flag to indicate that the image must be drawn slightly lighter

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.16 TVTPaintInfo Record

Not documented.

Pascal

```
TVTPaintInfo = record
  Canvas: TCanvas;
  PaintOptions: TVTInternalPaintOptions;
  Node: PVirtualNode;
  Column: TColumnIndex;
  Position: TColumnPosition;
  CellRect: TRect;
  ContentRect: TRect;
  NodeWidth: Integer;
  Alignment: TAlignment;
  BidiMode: TBidiMode;
  BrushOrigin: TPoint;
  ImageInfo: array[TVTImageInfoIndex] of TVTImageInfo;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
`Canvas: TCanvas;`	the canvas to paint on
`PaintOptions: TVTInternalPaintOptions;`	a copy of the paint options passed to PaintTree
`Node: PVirtualNode;`	the node to paint
`Column: TColumnIndex;`	the node's column index to paint
`Position: TColumnPosition;`	the column position of the node
`NodeWidth: Integer;`	the actual node width
`Alignment: TAlignment;`	how to align within the node rectangle
`BidiMode: TBidiMode;`	directionality to be used for painting
`BrushOrigin: TPoint;`	the alignment for the brush used to draw dotted lines
`ImageInfo: array[TVTImageInfoIndex] of TVTImageInfo;`	info about each possible node image

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.17 TVTReference Record

Not documented.

Pascal

```
TVTReference = record
  Process: Cardinal;
  Tree: TBaseVirtualTree;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.18 TVTTooltipLineBreakStyle Enumeration

Not documented.

Pascal

```
TVTTooltipLineBreakStyle = (hlbDefault, hlbForceSingleLine, hlbForceMultiLine);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.3.19 TWMPrint Record

Not documented.

Pascal

```
TWMPrint = packed record
  Msg: Cardinal;
  DC: HDC;
  Flags: Cardinal;
  Result: Integer;
  end;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Structs and Records(see page 569)

File

VirtualTrees

10.4 Types

These are all types that are contained in this documentation.

Enumerations

TAddPopupItemType(see TAddPopupItemType Enumeration, page 630)
Not documented.

TBlendMode(see TBlendMode Enumeration, page 631)
Not documented.

TChangeReason(see TChangeReason Enumeration, page 631)
Not documented.

TCheckImageKind(see TCheckImageKind Enumeration, page 632)
Determines which images should be used for checkboxes and radio buttons.

TCheckState(see TCheckState Enumeration, page 634)
Returns the current state of a node's check box, radio button or node button.

TCheckType(see TCheckType Enumeration, page 634)
Not documented.

TDragOperation(see TDragOperation Enumeration, page 635)
Not documented.

TDropMode(see TDropMode Enumeration, page 635)
Not documented.

THeaderState(see THeaderState Enumeration, page 636)
Not documented.

THintAnimationType(see THintAnimationType Enumeration, page 636)
Not documented.

THitPosition(see THitPosition Enumeration, page 637)
Not documented.

TItemEraseAction(see TItemEraseAction Enumeration, page 637)
Not documented.

TScrollBarStyle(see TScrollBarStyle Enumeration, page 638)
Not documented.

TSortDirection(see TSortDirection Enumeration, page 638)
Not documented.

TVirtualNodeInitState(see TVirtualNodeInitState Enumeration, page 638)
Not documented.

TVirtualNodeState(see TVirtualNodeState Enumeration, page 639)
Not documented.

TVirtualTreeColumnStyle(see TVirtualTreeColumnStyle Enumeration, page 640)
Not documented.

TVSTTextSourceType(see TVSTTextSourceType Enumeration, page 640)
Not documented.

TVSTTextType(see TVSTTextType Enumeration, page 640)
Not documented.

TVTAnimationOption(see TVTAnimationOption Enumeration, page 641)
Not documented.

TVTAutoOption(see TVTAutoOption Enumeration, page 641)
Not documented.

TVTButtonFillMode(see TVTButtonFillMode Enumeration, page 642)

Determines how the interior of nodes buttons should be drawn.

TVTButtonStyle(see TVTButtonStyle Enumeration, page 643)
Not documented.

TVTColumnOption(see TVTColumnOption Enumeration, page 643)
Not documented.

TVTDragImageKind(see TVTDragImageKind Enumeration, page 644)
Not documented.

TVTDragMoveRestriction(see TVTDragMoveRestriction Enumeration, page 644)
Not documented.

TVTDragType(see TVTDragType Enumeration, page 644)
Not documented.

TVTDrawSelectionMode(see TVTDrawSelectionMode Enumeration, page 645)
Not documented.

TVTDropMarkMode(see TVTDropMarkMode Enumeration, page 645)
Not documented.

TVTHeaderColumnLayout(see TVTHeaderColumnLayout Enumeration, page 645)
Not documented.

TVTHeaderOption(see TVTHeaderOption Enumeration, page 646)
Not documented.

TVTHeaderPopupOption(see TVTHeaderPopupOption Enumeration, page 646)
Not documented.

TVTHeaderStyle(see TVTHeaderStyle Enumeration, page 647)
Not documented.

TVTHintMode(see TVTHintMode Enumeration, page 648)
Not documented.

TVTImageInfoIndex(see TVTImageInfoIndex Enumeration, page 648)
Not documented.

TVTImageKind(see TVTImageKind Enumeration, page 648)
Not documented.

TVTIncrementalSearch(see TVTIncrementalSearch Enumeration, page 649)
Not documented.

TVTInternalPaintOption(see TVTInternalPaintOption Enumeration, page 649)
Not documented.

TVTLineMode(see TVTLineMode Enumeration, page 650)
Not documented.

TVTLineStyle(see TVTLineStyle Enumeration, page 650)
Not documented.

TVTLineType(see TVTLineType Enumeration, page 651)
Not documented.

TVTMiscOption(see TVTMiscOption Enumeration, page 651)
Not documented.

TVTNodeAlignment(see TVTNodeAlignment Enumeration, page 652)
Not documented.

TVTNodeAttachMode(see TVTNodeAttachMode Enumeration, page 652)
Not documented.

TVTPaintOption(see TVTPaintOption Enumeration, page 653)
Not documented.

TVTSearchDirection(see TVTSearchDirection Enumeration, page 654)
Not documented.

TVTSearchStart(see TVTSearchStart Enumeration, page 654)
Not documented.

TVTSelectionOption(see TVTSelectionOption Enumeration, page 655)
Not documented.

TVTStringOption(see TVTStringOption Enumeration, page 655)
Not documented.

TVTUpdateState(see TVTUpdateState Enumeration, page 656)
Not documented.

Group

Virtual Treeview(see page 71)

Types

PCardinal(see PCardinal Type, page 587)
Not documented.

PClipboardFormatListEntry(see PClipboardFormatListEntry Type, page 587)
Not documented.

PSHDragImage(see PSHDragImage Type, page 588)
Not documented.

PVirtualNode(see PVirtualNode Type, page 588)
Not documented.

PVTHintData(see PVTHintData Type, page 588)
Not documented.

PVTReference(see PVTReference Type, page 589)
Not documented.

TAddHeaderPopupItemEvent(see TAddHeaderPopupItemEvent Type, page 589)
Not documented.

TAutoScrollInterval(see TAutoScrollInterval Type, page 589)
Not documented.

TCache(see TCache Type, page 590)
Not documented.

TCardinalArray(see TCardinalArray Type, page 590)
Not documented.

TChangeStates(see TChangeStates Type, page 590)
Not documented.

TColumnChangeEvent(see TColumnChangeEvent Type, page 591)
Not documented.

TColumnIndex(see TColumnIndex Type, page 591)
Not documented.

TColumnPosition(see TColumnPosition Type, page 591)
Not documented.

TColumnsArray(see TColumnsArray Type, page 592)
Not documented.

TDragOperations(see TDragOperations Type, page 592)
Not documented.

TFormatArray(see TFormatArray Type, page 592)
Not documented.

TFormatEtcArray(see TFormatEtcArray Type, page 593)
Not documented.

TGetFirstNodeProc(see TGetFirstNodeProc Type, page 593)
Not documented.

TGetNextNodeProc(see TGetNextNodeProc Type, page 593)
Not documented.

THeaderPaintElements(see THeaderPaintElements Type, page 594)

Not documented.

THeaderStates(see THeaderStates Type, page 594)
Not documented.

THitPositions(see THitPositions Type, page 594)
Not documented.

TImageIndex(see TImageIndex Type, page 595)
Not documented.

TIndexArray(see TIndexArray Type, page 595)
Not documented.

TInternalStgMediumArray(see TInternalStgMediumArray Type, page 595)
Not documented.

TLineImage(see TLineImage Type, page 596)
Not documented.

TMagicID(see TMagicID Type, page 596)
Not documented.

TMouseButtons(see TMouseButtons Type, page 596)
Not documented.

TNodeArray(see TNodeArray Type, page 597)
Not documented.

TScrollDirections(see TScrollDirections Type, page 597)
Not documented.

TScrollUpdateOptions(see TScrollUpdateOptions Type, page 597)
Not documented.

TTreeOptionsClass(see TTreeOptionsClass Type, page 598)
Not documented.

TVirtualNodeInitStates(see TVirtualNodeInitStates Type, page 598)
Not documented.

TVirtualNodeStates(see TVirtualNodeStates Type, page 598)
Not documented.

TVirtualTreeClass(see TVirtualTreeClass Type, page 599)
Not documented.

TVirtualTreeColumnClass(see TVirtualTreeColumnClass Type, page 599)
Not documented.

TVirtualTreeColumnsClass(see TVirtualTreeColumnsClass Type, page 599)
Not documented.

TVirtualTreeStates(see TVirtualTreeStates Type, page 600)
Not documented.

TVSTGetTextEvent(see TVSTGetTextEvent Type, page 600)
Not documented.

TVSTNewTextEvent(see TVSTNewTextEvent Type, page 601)
Not documented.

TVSTShortenStringEvent(see TVSTShortenStringEvent Type, page 601)
Not documented.

TVTAdvancedHeaderPaintEvent(see TVTAdvancedHeaderPaintEvent Type, page 601)
Not documented.

TVTAfterCellPaintEvent(see TVTAfterCellPaintEvent Type, page 602)
Not documented.

TVTAfterItemEraseEvent(see TVTAfterItemEraseEvent Type, page 602)
Not documented.

TVTAfterItemPaintEvent(see TVTAfterItemPaintEvent Type, page 602)
Not documented.

TVTAnimationCallback(see TVTAnimationCallback Type, page 603)
Not documented.

TVTAnimationOptions(see TVTAnimationOptions Type, page 603)
Not documented.

TVTAutoOptions(see TVTAutoOptions Type, page 603)
Not documented.

TVTBackgroundPaintEvent(see TVTBackgroundPaintEvent Type, page 604)
Not documented.

TVTBeforeCellPaintEvent(see TVTBeforeCellPaintEvent Type, page 604)
Not documented.

TVTBeforeItemEraseEvent(see TVTBeforeItemEraseEvent Type, page 604)
Not documented.

TVTBeforeItemPaintEvent(see TVTBeforeItemPaintEvent Type, page 605)
Not documented.

TVTBias(see TVTBias Type, page 605)
Not documented.

TVTChangeEvent(see TVTChangeEvent Type, page 605)
Not documented.

TVTChangingEvent(see TVTChangingEvent Type, page 606)
Not documented.

TVTCheckChangingEvent(see TVTCheckChangingEvent Type, page 606)
Not documented.

TVTColumnClickEvent(see TVTColumnClickEvent Type, page 606)
Not documented.

TVTColumnDblClickEvent(see TVTColumnDblClickEvent Type, page 607)
Not documented.

TVTColumnOptions(see TVTColumnOptions Type, page 607)
Not documented.

TVTCompareEvent(see TVTCompareEvent Type, page 607)
Not documented.

TVTCreateDataObjectEvent(see TVTCreateDataObjectEvent Type, page 608)
Not documented.

TVTCreateDragManagerEvent(see TVTCreateDragManagerEvent Type, page 608)
Not documented.

TVTCreateEditorEvent(see TVTCreateEditorEvent Type, page 609)
Not documented.

TVTDragAllowedEvent(see TVTDragAllowedEvent Type, page 609)
Not documented.

TVTDragDropEvent(see TVTDragDropEvent Type, page 609)
Not documented.

TVTDragImageStates(see TVTDragImageStates Type, page 610)
Not documented.

TVTDragOverEvent(see TVTDragOverEvent Type, page 610)
Not documented.

TVTDrawHintEvent(see TVTDrawHintEvent Type, page 610)
Not documented.

TVTDrawNodeEvent(see TVTDrawNodeEvent Type, page 611)
Not documented.

TVTEditCancelEvent(see TVTEditCancelEvent Type, page 611)
Not documented.

TVTEditChangeEvent(see TVTEditChangeEvent Type, page 611)

Not documented.

TVTEditChangingEvent(see TVTEditChangingEvent Type, page 612)
Not documented.

TVTFocusChangeEvent(see TVTFocusChangeEvent Type, page 612)
Not documented.

TVTFocusChangingEvent(see TVTFocusChangingEvent Type, page 612)
Not documented.

TVTFreeNodeEvent(see TVTFreeNodeEvent Type, page 613)
Not documented.

TVTGetCursorEvent(see TVTGetCursorEvent Type, page 613)
Not documented.

TVTGetHeaderCursorEvent(see TVTGetHeaderCursorEvent Type, page 613)
Not documented.

TVTGetHintSizeEvent(see TVTGetHintSizeEvent Type, page 614)
Not documented.

TVTGetImageEvent(see TVTGetImageEvent Type, page 614)
Not documented.

TVTGetLineStyleEvent(see TVTGetLineStyleEvent Type, page 614)
Not documented.

TVTGetNodeDataSizeEvent(see TVTGetNodeDataSizeEvent Type, page 615)
Not documented.

TVTGetNodeProc(see TVTGetNodeProc Type, page 615)
Not documented.

TVTGetNodeWidthEvent(see TVTGetNodeWidthEvent Type, page 615)
Not documented.

TVTGetUserClipboardFormatsEvent(see TVTGetUserClipboardFormatsEvent Type, page 616)
Not documented.

TVTHeaderClass(see TVTHeaderClass Type, page 616)
Not documented.

TVTHeaderClickEvent(see TVTHeaderClickEvent Type, page 616)
Not documented.

TVTHeaderDraggedEvent(see TVTHeaderDraggedEvent Type, page 617)
Not documented.

TVTHeaderDraggedOutEvent(see TVTHeaderDraggedOutEvent Type, page 617)
Not documented.

TVSTGetHintEvent(see TVSTGetHintEvent Type, page 617)
Not documented.

TVTHeaderDraggingEvent(see TVTHeaderDraggingEvent Type, page 618)
Not documented.

TVTHeaderMouseEvent(see TVTHeaderMouseEvent Type, page 618)
Not documented.

TVTHeaderMouseMoveEvent(see TVTHeaderMouseMoveEvent Type, page 618)
Not documented.

TVTHeaderNotifyEvent(see TVTHeaderNotifyEvent Type, page 619)
Not documented.

TVTHeaderOptions(see TVTHeaderOptions Type, page 619)
Not documented.

TVTHeaderPaintEvent(see TVTHeaderPaintEvent Type, page 619)
Not documented.

TVTHeaderPaintQueryElementsEvent(see TVTHeaderPaintQueryElementsEvent Type, page 620)
Not documented.

TVTHeaderPopupOptions(see TVTHeaderPopupOptions Type, page 620)
Not documented.

TVTHelpContextEvent(see TVTHelpContextEvent Type, page 620)
Not documented.

TVTHotNodeChangeEvent(see TVTHotNodeChangeEvent Type, page 621)
Not documented.

TVTIncrementalSearchEvent(see TVTIncrementalSearchEvent Type, page 621)
Not documented.

TVTInitChildrenEvent(see TVTInitChildrenEvent Type, page 621)
Not documented.

TVTInitNodeEvent(see TVTInitNodeEvent Type, page 622)
Not documented.

TVTInternalPaintOptions(see TVTInternalPaintOptions Type, page 622)
Not documented.

TVTKeyActionEvent(see TVTKeyActionEvent Type, page 622)
Not documented.

TVTMeasureItemEvent(see TVTMeasureItemEvent Type, page 623)
Not documented.

TVTMiscOptions(see TVTMiscOptions Type, page 623)
Not documented.

TVTNodeCopiedEvent(see TVTNodeCopiedEvent Type, page 623)
Not documented.

TVTNodeCopyingEvent(see TVTNodeCopyingEvent Type, page 624)
Not documented.

TVTNodeMovedEvent(see TVTNodeMovedEvent Type, page 624)
Not documented.

TVTNodeMovingEvent(see TVTNodeMovingEvent Type, page 624)
Not documented.

TVTPaintEvent(see TVTPaintEvent Type, page 625)
Not documented.

TVTPaintOptions(see TVTPaintOptions Type, page 625)
Not documented.

TVTPaintText(see TVTPaintText Type, page 625)
Not documented.

TVTPopupEvent(see TVTPopupEvent Type, page 626)
Not documented.

TVTRenderOLEDataEvent(see TVTRenderOLEDataEvent Type, page 626)
Not documented.

TVTSaveNodeEvent(see TVTSaveNodeEvent Type, page 627)
Not documented.

TVTScrollEvent(see TVTScrollEvent Type, page 627)
Not documented.

TVTScrollIncrement(see TVTScrollIncrement Type, page 627)
Not documented.

TVTSelectionOptions(see TVTSelectionOptions Type, page 628)
Not documented.

TVTStateChangeEvent(see TVTStateChangeEvent Type, page 628)
Not documented.

TVTStringOptions(see TVTStringOptions Type, page 628)
Not documented.

TVTStructureChangeEvent(see TVTStructureChangeEvent Type, page 629)

Not documented.

TVTTransparency(see TVTTransparency Type, page 629)
Not documented.

TVTUpdatingEvent(see TVTUpdatingEvent Type, page 629)
Not documented.

TWMContextMenu(see TWMContextMenu Type, page 630)
Not documented.

TWMPrintClient(see TWMPrintClient Type, page 630)
Not documented.

TVTGetCellIsEmptyEvent(see TVTGetCellIsEmptyEvent Type, page 631)
Not documented.

TVTGetImageExEvent(see TVTGetImageExEvent Type, page 635)
Not documented.

TVTMenuItem(see TVTMenuItem Type, page 647)
Not documented.

TVTScrollbarShowEvent(see TVTScrollbarShowEvent Type, page 653)
Not documented.

Legend

Type

Struct

10.4.1 PCardinal Type

Not documented.

Pascal

```
PCardinal = ^Cardinal;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.2 PClipboardFormatListEntry Type

Not documented.

Pascal

```
PClipboardFormatListEntry = ^TClipboardFormatListEntry;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.3 PSHDragImage Type

Not documented.

Pascal

```
PSHDragImage = ^TSHDragImage;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.4 PVirtualNode Type

Not documented.

Pascal

```
PVirtualNode = ^TVirtualNode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.5 PVTHintData Type

Not documented.

Pascal

```
PVTHintData = ^TVTHintData;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.6 PVTReference Type

Not documented.

Pascal

```
PVTReference = ^TVTReference;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.7 TAddHeaderPopupItemEvent Type

Not documented.

Pascal

```
TAddHeaderPopupItemEvent = procedure (const Sender: TBaseVirtualTree; const Column: TColumnIndex; var
Cmd: TAddPopupItemType) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VTHeaderPopup

10.4.8 TAutoScrollInterval Type

Not documented.

Pascal

```
TAutoScrollInterval = 1..1000;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.9 TCache Type

Not documented.

Pascal

```
TCache = array of TCacheEntry;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.10 TCardinalArray Type

Not documented.

Pascal

```
TCardinalArray = array of Cardinal;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.11 TChangeStates Type

Not documented.

Pascal

```
TChangeStates = set of ( csStopValidation, csUseCache, csValidating, csValidationNeeded );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.12 TColumnChangeEvent Type

Not documented.

Pascal

```
TColumnChangeEvent = procedure (const Sender: TBaseVirtualTree; const Column: TColumnIndex; Visible:
Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VTHeaderPopup

10.4.13 TColumnIndex Type

Not documented.

Pascal

```
TColumnIndex = type Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.14 TColumnPosition Type

Not documented.

Pascal

```
TColumnPosition = type Cardinal;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.15 TColumnsArray Type

Not documented.

Pascal

```
TColumnsArray = array of TVirtualTreeColumn;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.16 TDragOperations Type

Not documented.

Pascal

```
TDragOperations = set of TDragOperation;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.17 TFormatArray Type

Not documented.

Pascal

```
TFormatArray = array of Word;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.18 TFormatEtcArray Type

Not documented.

Pascal

```
TFormatEtcArray = array of TFormatEtc;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.19 TGetFirstNodeProc Type

Not documented.

Pascal

```
TGetFirstNodeProc = function : PVirtualNode of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.20 TGetNextNodeProc Type

Not documented.

Pascal

```
TGetNextNodeProc = function (Node: PVirtualNode): PVirtualNode of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.21 THeaderPaintElements Type

Not documented.

Pascal

```
THeaderPaintElements = set of ( hpeBackground, hpeDropMark, hpeHeaderGlyph, hpeSortGlyph, hpeText );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.22 THeaderStates Type

Not documented.

Pascal

```
THeaderStates = set of THeaderState;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.23 THitPositions Type

Not documented.

Pascal

```
THitPositions = set of THitPosition;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.24 TImageIndex Type

Not documented.

Pascal

```
TImageIndex = Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.25 TIndexArray Type

Not documented.

Pascal

```
TIndexArray = array of TColumnIndex;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.26 TInternalStgMediumArray Type

Not documented.

Pascal

```
TInternalStgMediumArray = array of TInternalStgMedium;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.27 TLineImage Type

Not documented.

Pascal

```
TLineImage = array of TVTLineType;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.28 TMagicID Type

Not documented.

Pascal

```
TMagicID = array[0..5] of WideChar;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.29 TMouseButtons Type

Not documented.

Pascal

```
TMouseButtons = set of TMouseButton;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.30 TNodeArray Type

Not documented.

Pascal

```
TNodeArray = array of PVirtualNode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.31 TScrollDirections Type

Not documented.

Pascal

```
TScrollDirections = set of ( sdLeft, sdUp, sdRight, sdDown );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.32 TScrollUpdateOptions Type

Not documented.

Pascal

```
TScrollUpdateOptions = set of ( suoRepaintHeader, suoRepaintScrollbars, suoScrollClientArea,
suoUpdateNCArea );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.33 TTreeOptionsClass Type

Not documented.

Pascal

```
TTreeOptionsClass = class of TCustomVirtualTreeOptions;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.34 TVirtualNodeInitStates Type

Not documented.

Pascal

```
TVirtualNodeInitStates = set of TVirtualNodeInitState;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.35 TVirtualNodeStates Type

Not documented.

Pascal

```
TVirtualNodeStates = set of TVirtualNodeState;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.36 TVirtualTreeClass Type

Not documented.

Pascal

```
TVirtualTreeClass = class of TBaseVirtualTree;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.37 TVirtualTreeColumnClass Type

Not documented.

Pascal

```
TVirtualTreeColumnClass = class of TVirtualTreeColumn;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.38 TVirtualTreeColumnsClass Type

Not documented.

Pascal

```
TVirtualTreeColumnsClass = class of TVirtualTreeColumns;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.39 TVirtualTreeStates Type

Not documented.

Pascal

```
TVirtualTreeStates = set of ( tsCancelHintAnimation, tsChangePending, tsCheckPropagation,
tsCollapsing, tsToggleFocusedSelection, tsClearPending, tsClipboardFlushing, tsCopyPending,
tsCutPending, tsDrawSelPending, tsDrawSelecting, tsEditing, tsEditPending, tsExpanding, tsHint,
tsInAnimation, tsIncrementalSearching, tsIncrementalSearchPending, tsIterating, tsKeyCheckPending,
tsLeftButtonDown, tsMouseCheckPending, tsMiddleButtonDown, tsNeedScale, tsNeedRootCountUpdate,
tsOLEDragging, tsOLEDragPending, tsPainting, tsRightButtonDown, tsPopupMenuShown, tsScrolling,
tsScrollPending, tsSizing, tsStopValidation, tsStructureChangePending, tsSynchMode, tsThumbTracking,
tsUpdateHiddenChildrenNeeded, tsUpdating, tsUseCache, tsUserDragObject, tsUseThemes, tsValidating,
tsValidationNeeded, tsVCLDragging, tsVCLDragPending, tsWheelPanning, tsWheelScrolling,
tsWindowCreating );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.40 TVSTGetTextEvent Type

Not documented.

Pascal

```
TVSTGetTextEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex;
TextType: TVSTTextType; var CellText: WideString) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.41 TVSTNewTextEvent Type

Not documented.

Pascal

```
TVSTNewTextEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex;
NewText: WideString) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.42 TVSTShortenStringEvent Type

Not documented.

Pascal

```
TVSTShortenStringEvent = procedure (Sender: TBaseVirtualTree; TargetCanvas: TCanvas; Node:
PVirtualNode; Column: TColumnIndex; const S: WideString; TextSpace: Integer; RightToLeft: Boolean; var
Result: WideString; var Done: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.43 TVTAdvancedHeaderPaintEvent Type

Not documented.

Pascal

```
TVTAdvancedHeaderPaintEvent = procedure (Sender: TVTHeader; var PaintInfo: THeaderPaintInfo; const
Elements: THeaderPaintElements) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.44 TVTAfterCellPaintEvent Type

Not documented.

Pascal

```
TVTAfterCellPaintEvent = procedure (Sender: TBaseVirtualTree; TargetCanvas: TCanvas; Node:
PVirtualNode; Column: TColumnIndex; CellRect: TRect) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.45 TVTAfterItemEraseEvent Type

Not documented.

Pascal

```
TVTAfterItemEraseEvent = procedure (Sender: TBaseVirtualTree; TargetCanvas: TCanvas; Node:
PVirtualNode; ItemRect: TRect) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.46 TVTAfterItemPaintEvent Type

Not documented.

Pascal

```
TVTAfterItemPaintEvent = procedure (Sender: TBaseVirtualTree; TargetCanvas: TCanvas; Node:
PVirtualNode; ItemRect: TRect) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.47 TVTAnimationCallback Type

Not documented.

Pascal

```
TVTAnimationCallback = function (Step, StepSize: Integer; Data: Pointer): Boolean of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.48 TVTAnimationOptions Type

Not documented.

Pascal

```
TVTAnimationOptions = set of TVTAnimationOption;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.49 TVTAutoOptions Type

Not documented.

Pascal

```
TVTAutoOptions = set of TVTAutoOption;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.50 TVTBackgroundPaintEvent Type

Not documented.

Pascal

```
TVTBackgroundPaintEvent = procedure (Sender: TBaseVirtualTree; TargetCanvas: TCanvas; R: TRect; var
Handled: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.51 TVTBeforeCellPaintEvent Type

Not documented.

Pascal

```
TVTBeforeCellPaintEvent = procedure (Sender: TBaseVirtualTree; TargetCanvas: TCanvas; Node:
PVirtualNode; Column: TColumnIndex; CellRect: TRect) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.52 TVTBeforeItemEraseEvent Type

Not documented.

Pascal

```
TVTBeforeItemEraseEvent = procedure (Sender: TBaseVirtualTree; TargetCanvas: TCanvas; Node:
PVirtualNode; ItemRect: TRect; var ItemColor: TColor; var EraseAction: TItemEraseAction) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.53 TVTBeforeItemPaintEvent Type

Not documented.

Pascal

```
TVTBeforeItemPaintEvent = procedure (Sender: TBaseVirtualTree; TargetCanvas: TCanvas; Node:
PVirtualNode; ItemRect: TRect; var CustomDraw: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.54 TVTBias Type

Not documented.

Pascal

```
TVTBias = -128..127;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.55 TVTChangeEvent Type

Not documented.

Pascal

```
TVTChangeEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.56 TVTChangingEvent Type

Not documented.

Pascal

```
TVTChangingEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; var Allowed: Boolean) of
object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.57 TVTCheckChangingEvent Type

Not documented.

Pascal

```
TVTCheckChangingEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; var NewState:
TCheckState; var Allowed: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.58 TVTColumnClickEvent Type

Not documented.

Pascal

```
TVTColumnClickEvent = procedure (Sender: TBaseVirtualTree; Column: TColumnIndex; Shift: TShiftState)
of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.59 TVTColumnDblClickEvent Type

Not documented.

Pascal

```
TVTColumnDblClickEvent = procedure (Sender: TBaseVirtualTree; Column: TColumnIndex; Shift:
TShiftState) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.60 TVTColumnOptions Type

Not documented.

Pascal

```
TVTColumnOptions = set of TVTColumnOption;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.61 TVTCompareEvent Type

Not documented.

Pascal

```
TVTCompareEvent = procedure (Sender: TBaseVirtualTree; Node1, Node2: PVirtualNode; Column:
TColumnIndex; var Result: Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.62 TVTCreateDataObjectEvent Type

Not documented.

Pascal

```
TVTCreateDataObjectEvent = procedure (Sender: TBaseVirtualTree; out IDataObject: IDataObject) of
object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.63 TVTCreateDragManagerEvent Type

Not documented.

Pascal

```
TVTCreateDragManagerEvent = procedure (Sender: TBaseVirtualTree; out DragManager: IVTDragManager) of
object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.64 TVTCreateEditorEvent Type

Not documented.

Pascal

```
TVTCreateEditorEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex;
out EditLink: IVTEditLink) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.65 TVTDragAllowedEvent Type

Not documented.

Pascal

```
TVTDragAllowedEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex;
var Allowed: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.66 TVTDragDropEvent Type

Not documented.

Pascal

```
TVTDragDropEvent = procedure (Sender: TBaseVirtualTree; Source: TObject; DataObject: IDataObject;
Formats: TFormatArray; Shift: TShiftState; Pt: TPoint; var Effect: Integer; Mode: TDropMode) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.67 TVTDragImageStates Type

Not documented.

Pascal

```
TVTDragImageStates = set of ( disHidden, disInDrag, disPrepared, disSystemSupport );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.68 TVTDragOverEvent Type

Not documented.

Pascal

```
TVTDragOverEvent = procedure (Sender: TBaseVirtualTree; Source: TObject; Shift: TShiftState; State:
TDragState; Pt: TPoint; Mode: TDropMode; var Effect: Integer; var Accept: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.69 TVTDrawHintEvent Type

Not documented.

Pascal

```
TVTDrawHintEvent = procedure (Sender: TBaseVirtualTree; HintCanvas: TCanvas; Node: PVirtualNode; R:
TRect; Column: TColumnIndex) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.70 TVTDrawNodeEvent Type

Not documented.

Pascal

```
TVTDrawNodeEvent = procedure (Sender: TBaseVirtualTree; const PaintInfo: TVTPaintInfo) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.71 TVTEditCancelEvent Type

Not documented.

Pascal

```
TVTEditCancelEvent = procedure (Sender: TBaseVirtualTree; Column: TColumnIndex) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.72 TVTEditChangeEvent Type

Not documented.

Pascal

```
TVTEditChangeEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex) of
object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.73 TVTEditChangingEvent Type

Not documented.

Pascal

```
TVTEditChangingEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex;
var Allowed: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.74 TVTFocusChangeEvent Type

Not documented.

Pascal

```
TVTFocusChangeEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex)
of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.75 TVTFocusChangingEvent Type

Not documented.

Pascal

```
TVTFocusChangingEvent = procedure (Sender: TBaseVirtualTree; OldNode, NewNode: PVirtualNode;
OldColumn, NewColumn: TColumnIndex; var Allowed: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.76 TVTFreeNodeEvent Type

Not documented.

Pascal

```
TVTFreeNodeEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.77 TVTGetCursorEvent Type

Not documented.

Pascal

```
TVTGetCursorEvent = procedure (Sender: TBaseVirtualTree; var Cursor: TCursor) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.78 TVTGetHeaderCursorEvent Type

Not documented.

Pascal

```
TVTGetHeaderCursorEvent = procedure (Sender: TVTHeader; var Cursor: HCURSOR) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.79 TVTGetHintSizeEvent Type

Not documented.

Pascal

```
TVTGetHintSizeEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex;
var R: TRect) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.80 TVTGetImageEvent Type

Not documented.

Pascal

```
TVTGetImageEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Kind: TVTImageKind;
Column: TColumnIndex; var Ghosted: Boolean; var ImageIndex: Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.81 TVTGetLineStyleEvent Type

Not documented.

Pascal

```
TVTGetLineStyleEvent = procedure (Sender: TBaseVirtualTree; var Bits: Pointer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.82 TVTGetNodeDataSizeEvent Type

Not documented.

Pascal

```
TVTGetNodeDataSizeEvent = procedure (Sender: TBaseVirtualTree; var NodeDataSize: Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.83 TVTGetNodeProc Type

Not documented.

Pascal

```
TVTGetNodeProc = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Data: Pointer; var Abort:
Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.84 TVTGetNodeWidthEvent Type

Not documented.

Pascal

```
TVTGetNodeWidthEvent = procedure (Sender: TBaseVirtualTree; HintCanvas: TCanvas; Node: PVirtualNode;
Column: TColumnIndex; var NodeWidth: Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.85 TVTGetUserClipboardFormatsEvent Type

Not documented.

Pascal

```
TVTGetUserClipboardFormatsEvent = procedure (Sender: TBaseVirtualTree; var Formats: TFormatEtcArray)
of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.86 TVTHeaderClass Type

Not documented.

Pascal

```
TVTHeaderClass = class of TVTHeader;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.87 TVTHeaderClickEvent Type

Not documented.

Pascal

```
TVTHeaderClickEvent = procedure (Sender: TVTHeader; Column: TColumnIndex; Button: TMouseButton; Shift:
TShiftState; X, Y: Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.88 TVTHeaderDraggedEvent Type

Not documented.

Pascal

```
TVTHeaderDraggedEvent = procedure (Sender: TVTHeader; Column: TColumnIndex; OldPosition: Integer) of
object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.89 TVTHeaderDraggedOutEvent Type

Not documented.

Pascal

```
TVTHeaderDraggedOutEvent = procedure (Sender: TVTHeader; Column: TColumnIndex; DropPosition: TPoint)
of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.90 TVSTGetHintEvent Type

Not documented.

Pascal

```
TVSTGetHintEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex; var
LineBreakStyle: TVTTooltipLineBreakStyle; var HintText: WideString) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.91 TVTHeaderDraggingEvent Type

Not documented.

Pascal

```
TVTHeaderDraggingEvent = procedure (Sender: TVTHeader; Column: TColumnIndex; var Allowed: Boolean) of
object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.92 TVTHeaderMouseEvent Type

Not documented.

Pascal

```
TVTHeaderMouseEvent = procedure (Sender: TVTHeader; Button: TMouseButton; Shift: TShiftState; X, Y:
Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.93 TVTHeaderMouseMoveEvent Type

Not documented.

Pascal

```
TVTHeaderMouseMoveEvent = procedure (Sender: TVTHeader; Shift: TShiftState; X, Y: Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.94 TVTHeaderNotifyEvent Type

Not documented.

Pascal

```
TVTHeaderNotifyEvent = procedure (Sender: TVTHeader; Column: TColumnIndex) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.95 TVTHeaderOptions Type

Not documented.

Pascal

```
TVTHeaderOptions = set of TVTHeaderOption;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.96 TVTHeaderPaintEvent Type

Not documented.

Pascal

```
TVTHeaderPaintEvent = procedure (Sender: TVTHeader; HeaderCanvas: TCanvas; Column: TVirtualTreeColumn;
R: TRect; Hover, Pressed: Boolean; DropMark: TVTDropMarkMode) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.97 TVTHeaderPaintQueryElementsEvent Type

Not documented.

Pascal

```
TVTHeaderPaintQueryElementsEvent = procedure (Sender: TVTHeader; var PaintInfo: THeaderPaintInfo; var
Elements: THeaderPaintElements) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.98 TVTHeaderPopupOptions Type

Not documented.

Pascal

```
TVTHeaderPopupOptions = set of TVTHeaderPopupOption;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VTHeaderPopup

10.4.99 TVTHelpContextEvent Type

Not documented.

Pascal

```
TVTHelpContextEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex;
var HelpContext: Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.100 TVTHotNodeChangeEvent Type

Not documented.

Pascal

```
TVTHotNodeChangeEvent = procedure (Sender: TBaseVirtualTree; OldNode, NewNode: PVirtualNode) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.101 TVTIncrementalSearchEvent Type

Not documented.

Pascal

```
TVTIncrementalSearchEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; const SearchText:
WideString; var Result: Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.102 TVTInitChildrenEvent Type

Not documented.

Pascal

```
TVTInitChildrenEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; var ChildCount:
Cardinal) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.103 TVTInitNodeEvent Type

Not documented.

Pascal

```
TVTInitNodeEvent = procedure (Sender: TBaseVirtualTree; ParentNode, Node: PVirtualNode; var
InitialStates: TVirtualNodeInitStates) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.104 TVTInternalPaintOptions Type

Not documented.

Pascal

```
TVTInternalPaintOptions = set of TVTInternalPaintOption;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.105 TVTKeyActionEvent Type

Not documented.

Pascal

```
TVTKeyActionEvent = procedure (Sender: TBaseVirtualTree; var CharCode: Word; var Shift: TShiftState;
var DoDefault: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.106 TVTMeasureItemEvent Type

Not documented.

Pascal

```
TVTMeasureItemEvent = procedure (Sender: TBaseVirtualTree; TargetCanvas: TCanvas; Node: PVirtualNode;
var NodeHeight: Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.107 TVTMiscOptions Type

Not documented.

Pascal

```
TVTMiscOptions = set of TVTMiscOption;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.108 TVTNodeCopiedEvent Type

Not documented.

Pascal

```
TVTNodeCopiedEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.109 TVTNodeCopyingEvent Type

Not documented.

Pascal

```
TVTNodeCopyingEvent = procedure (Sender: TBaseVirtualTree; Node, Target: PVirtualNode; var Allowed:
Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.110 TVTNodeMovedEvent Type

Not documented.

Pascal

```
TVTNodeMovedEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.111 TVTNodeMovingEvent Type

Not documented.

Pascal

```
TVTNodeMovingEvent = procedure (Sender: TBaseVirtualTree; Node, Target: PVirtualNode; var Allowed:
Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.112 TVTPaintEvent Type

Not documented.

Pascal

```
TVTPaintEvent = procedure (Sender: TBaseVirtualTree; TargetCanvas: TCanvas) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.113 TVTPaintOptions Type

Not documented.

Pascal

```
TVTPaintOptions = set of TVTPaintOption;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.114 TVTPaintText Type

Not documented.

Pascal

```
TVTPaintText = procedure (Sender: TBaseVirtualTree; const TargetCanvas: TCanvas; Node: PVirtualNode;
Column: TColumnIndex; TextType: TVSTTextType) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.115 TVTPopupEvent Type

Not documented.

Pascal

```
TVTPopupEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex; const
P: TPoint; var AskParent: Boolean; var PopupMenu: TPopupMenu) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.116 TVTRenderOLEDataEvent Type

Not documented.

Pascal

```
TVTRenderOLEDataEvent = procedure (Sender: TBaseVirtualTree; const FormatEtcIn: TFormatEtc; out
Medium: TStgMedium; ForClipboard: Boolean; var Result: HRESULT) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.117 TVTSaveNodeEvent Type

Not documented.

Pascal

```
TVTSaveNodeEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Stream: TStream) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.118 TVTScrollEvent Type

Not documented.

Pascal

```
TVTScrollEvent = procedure (Sender: TBaseVirtualTree; DeltaX, DeltaY: Integer) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.119 TVTScrollIncrement Type

Not documented.

Pascal

```
TVTScrollIncrement = 1..10000;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.120 TVTSelectionOptions Type

Not documented.

Pascal

```
TVTSelectionOptions = set of TVTSelectionOption;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.121 TVTStateChangeEvent Type

Not documented.

Pascal

```
TVTStateChangeEvent = procedure (Sender: TBaseVirtualTree; Enter, Leave: TVirtualTreeStates) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.122 TVTStringOptions Type

Not documented.

Pascal

```
TVTStringOptions = set of TVTStringOption;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.123 TVTStructureChangeEvent Type

Not documented.

Pascal

```
TVTStructureChangeEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Reason:
TChangeReason) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.124 TVTTransparency Type

Not documented.

Pascal

```
TVTTransparency = 0..255;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.125 TVTUpdatingEvent Type

Not documented.

Pascal

```
TVTUpdatingEvent = procedure (Sender: TBaseVirtualTree; State: TVTUpdateState) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.126 TWMContextMenu Type

Not documented.

Pascal

```
TWMContextMenu = TWMMouse;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.127 TWMPrintClient Type

Not documented.

Pascal

```
TWMPrintClient = TWMPrint;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.128 TAddPopupItemType Enumeration

Not documented.

Pascal

```
TAddPopupItemType = (apNormal, apDisabled, apHidden);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VTHeaderPopup

10.4.129 TBlendMode Enumeration

Not documented.

Pascal

```
TBlendMode = (bmConstantAlpha, bmPerPixelAlpha, bmMasterAlpha, bmConstantAlphaAndColor);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
bmConstantAlpha	apply given constant alpha
bmPerPixelAlpha	use alpha value of the source pixel
bmMasterAlpha	use alpha value of source pixel and multiply it with the constant alpha value
bmConstantAlphaAndColor	blend the destination color with the given constant color und the constant alpha value

Group

Types(see page 580)

File

VirtualTrees

10.4.130 TVTGetCellIsEmptyEvent Type

Not documented.

Pascal

```
TVTGetCellIsEmptyEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Column:
TColumnIndex; var IsEmpty: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.131 TChangeReason Enumeration

Not documented.

Pascal

```
TChangeReason = (crIgnore, crAccumulated, crChildAdded, crChildDeleted, crNodeAdded, crNodeCopied,
crNodeMoved);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
crIgnore	used as placeholder
crAccumulated	used for delayed changes
crChildAdded	one or more child nodes have been added
crChildDeleted	one or more child nodes have been deleted
crNodeAdded	a node has been added
crNodeCopied	a node has been duplicated
crNodeMoved	a node has been moved to a new place

Group

Types(see page 580)

File

VirtualTrees

10.4.132 TCheckImageKind Enumeration

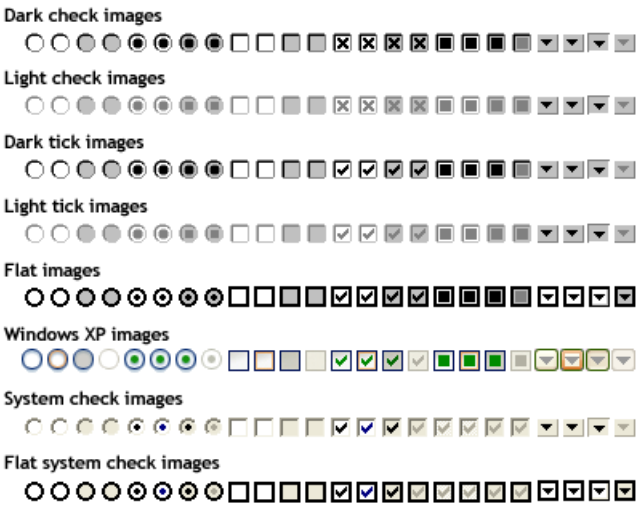
Determines which images should be used for checkboxes and radio buttons.

Pascal

```
TCheckImageKind = (ckLightCheck, ckDarkCheck, ckLightTick, ckDarkTick, ckFlat, ckXP, ckCustom,
ckSystem, ckSystemFlat);
```

Description

Provided with the tree are nine different image sets for the check images used when toCheckSupport is enabled in TreeOptions.



Eight of the nine lists are predefined while one is a custom check image list, which can be filled by the application. Use ckCustom as CheckImageKind value and assign an image list to the CustomCheckImages property to enable custom images.

The order of the images in the image lists is always as listed below. Make sure you have the same amount of images in your custom image list, if you want own check images.

- empty image (ckEmpty(see Check button image indices, page 674))

Radio buttons:

- uncheck normal (ckRadioUncheckedNormal(see Check button image indices, page 674))
- unchecked hot (ckRadioUncheckedHot(see Check button image indices, page 674))
- unchecked pressed (ckRadioUncheckedPressed(see Check button image indices, page 674))
- unchecked disabled (ckRadioUncheckedDisabled(see Check button image indices, page 674))
- checked normal (ckRadioCheckedNormal(see Check button image indices, page 674))
- checked hot (ckRadioCheckedHot(see Check button image indices, page 674))
- checked pressed (ckRadioCheckedPressed(see Check button image indices, page 674))
- checked disabled (ckRadioCheckedDisabled(see Check button image indices, page 674))

Check boxes:

- unchecked normal (ckCheckUncheckedNormal(see Check button image indices, page 674))
- unchecked hot (ckCheckUncheckedHot(see Check button image indices, page 674))
- unchecked pressed (ckCheckUncheckedPressed(see Check button image indices, page 674))
- unchecked disabled (ckCheckUncheckedDisabled(see Check button image indices, page 674))
- checked normal (ckCheckCheckedNormal(see Check button image indices, page 674))
- checked hot (ckCheckCheckedHot(see Check button image indices, page 674))
- checked pressed (ckCheckCheckedPressed(see Check button image indices, page 674))
- checked disabled (ckCheckCheckedDisabled(see Check button image indices, page 674))
- mixed normal (ckCheckMixedNormal(see Check button image indices, page 674))
- mixed hot (ckCheckMixedHot(see Check button image indices, page 674))
- mixed pressed (ckCheckMixedPressed(see Check button image indices, page 674))
- mixed disabled (ckCheckMixedDisabled(see Check button image indices, page 674))

Node buttons:

- button normal (ckButtonNormal(see Check button image indices, page 674))
- button hot (ckButtonHot(see Check button image indices, page 674))
- button pressed (ckButtonPressed(see Check button image indices, page 674))
- button disabled (ckButtonDisabled(see Check button image indices, page 674))

Members

Members	Description
ckLightCheck	gray cross
ckDarkCheck	black cross
ckLightTick	gray tick mark
ckDarkTick	black tick mark

ckFlat	flat images (no 3D border)
ckXP	Windows XP style
ckCustom	application defined check images
ckSystem	System defined check images.
ckSystemFlat	Flat system defined check images.

Group

Types(see page 580)

File

VirtualTrees

10.4.133 TCheckState Enumeration

Returns the current state of a node's check box, radio button or node button.

Pascal

```
TCheckState = (csUncheckedNormal, csUncheckedPressed, csCheckedNormal, csCheckedPressed,
csMixedNormal, csMixedPressed);
```

Description

The check states include both, transient and fluent (temporary) states. The only temporary state defined so far is the pressed state.

Members

Members	Description
csUncheckedNormal	unchecked and not pressed
csUncheckedPressed	unchecked and pressed
csCheckedNormal	checked and not pressed
csCheckedPressed	checked and pressed
csMixedNormal	3-state check box and not pressed
csMixedPressed	3-state check box and pressed

Group

Types(see page 580)

File

VirtualTrees

10.4.134 TCheckType Enumeration

Not documented.

Pascal

```
TCheckType = (ctNone, ctTriStateCheckBox, ctCheckBox, ctRadioButton, ctButton);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.135 TDragOperation Enumeration

Not documented.

Pascal

```
TDragOperation = (doCopy, doMove, doLink);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.136 TVTGetImageExEvent Type

Not documented.

Pascal

```
TVTGetImageExEvent = procedure (Sender: TBaseVirtualTree; Node: PVirtualNode; Kind: TVTImageKind;
Column: TColumnIndex; var Ghosted: Boolean; var ImageIndex: Integer; var ImageList: TCustomImageList)
of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.137 TDropMode Enumeration

Not documented.

Pascal

```
TDropMode = (dmNowhere, dmAbove, dmOnNode, dmBelow);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.138 THeaderState Enumeration

Not documented.

Pascal

```
THeaderState = (hsAutoSizing, hsDragging, hsDragPending, hsLoading, hsTracking, hsTrackPending);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
hsAutoSizing	auto size chain is in progess, do not trigger again on WM_SIZE
hsDragging	header dragging is in progress (only if enabled)
hsDragPending	left button is down, user might want to start dragging a column
hsLoading	The header currently loads from stream, so updates are not necessary.
hsTracking	column resizing is in progress
hsTrackPending	left button is down, user might want to start resize a column

Group

Types(see page 580)

File

VirtualTrees

10.4.139 THintAnimationType Enumeration

Not documented.

Pascal

```
THintAnimationType = (hatNone, hatFade, hatSlide, hatSystemDefault);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
hatNone	no animation at all, just display hint/tooltip
hatFade	fade in the hint/tooltip, like in Windows 2000
hatSlide	slide in the hint/tooltip, like in Windows 98
hatSystemDefault	use what the system is using (slide for Win9x, slide/fade for Win2K+, depends on settings)

Group

Types(see page 580)

File

VirtualTrees

10.4.140 THitPosition Enumeration

Not documented.

Pascal

```
THitPosition = (hiAbove, hiBelow, hiNowhere, hiOnItem, hiOnItemButton, hiOnItemCheckbox,
hiOnItemIndent, hiOnItemLabel, hiOnItemLeft, hiOnItemRight, hiOnNormalIcon, hiOnStateIcon, hiToLeft,
hiToRight);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
hiAbove	above the client area (if relative) or the absolute tree area
hiBelow	below the client area (if relative) or the absolute tree area
hiNowhere	no node is involved (possible only if the tree is not as tall as the client area)
hiOnItem	on the bitmaps/buttons or label associated with an item
hiOnItemButton	on the button associated with an item
hiOnItemCheckbox	on the checkbox if enabled
hiOnItemIndent	in the indentation area in front of a node
hiOnItemLabel	on the normal text area associated with an item
hiOnItemLeft	when right aligned or centered)
hiOnItemRight	if left aligned or centered)
hiOnNormalIcon	on the "normal" image
hiOnStateIcon	on the state image
hiToLeft	to the left of the client area (if relative) or the absolute tree area
hiToRight	to the right of the client area (if relative) or the absolute tree area

Group

Types(see page 580)

File

VirtualTrees

10.4.141 TItemEraseAction Enumeration

Not documented.

Pascal

```
TItemEraseAction = (eaColor, eaDefault, eaNone);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
eaColor	Use the provided color to erase the background instead the one of the tree.
eaDefault	The tree should erase the item's background (bitmap or solid).
eaNone	Let the application paint the background.

Group

Types(see page 580)

File

VirtualTrees

10.4.142 TScrollBarStyle Enumeration

Not documented.

Pascal

```
TScrollBarStyle = (sbmRegular, sbmFlat, sbm3D);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.143 TSortDirection Enumeration

Not documented.

Pascal

```
TSortDirection = (sdAscending, sdDescending);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.144 TVirtualNodeInitState Enumeration

Not documented.

Pascal

```
TVirtualNodeInitState = (ivsDisabled, ivsExpanded, ivsHasChildren, ivsMultiline, ivsSelected);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.145 TVirtualNodeState Enumeration

Not documented.

Pascal

```
TVirtualNodeState = (vsInitialized, vsChecking, vsCutOrCopy, vsDisabled, vsDeleting, vsExpanded,
vsHasChildren, vsVisible, vsSelected, vsInitialUserData, vsAllChildrenHidden, vsClearing, vsMultiline,
vsHeightMeasured, vsToggling);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
vsInitialized	Set after the node has been initialized.
vsChecking	Node's check state is changing, avoid propagation.
vsCutOrCopy	Node is selected as cut or copy and paste source.
vsDisabled	Set if node is disabled.
vsDeleting	Set when the node is about to be freed.
vsExpanded	Set if the node is expanded.
vsHasChildren	Indicates the presence of child nodes without actually setting them.
vsVisible	Indicate whether the node is visible or not (independant of the expand states of its parents).
vsSelected	Set if the node is in the current selection.
vsInitialUserData	Set if (via AddChild or InsertNode) initial user data has been set which requires OnFreeNode.
vsAllChildrenHidden	Set if vsHasChildren is set and no child node has the vsVisible flag set.
vsClearing	Don't register structure change event.
vsMultiline	Node text is wrapped at the cell boundaries instead of being shorted.
vsHeightMeasured	Node height has been determined and does not need a recalculation.

Group

Types(see page 580)

File

VirtualTrees

10.4.146 TVirtualTreeColumnStyle Enumeration

Not documented.

Pascal

```
TVirtualTreeColumnStyle = (vsText, vsOwnerDraw);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.147 TVSTTextSourceType Enumeration

Not documented.

Pascal

```
TVSTTextSourceType = (tstAll, tstInitialized, tstSelected, tstCutCopySet, tstVisible);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
tstAll	Initialization is done on the fly.
tstInitialized	Only initialized nodes are rendered.
tstSelected	Only selected nodes are rendered.
tstCutCopySet	Only nodes currently marked as being in the cut/copy clipboard set are rendered.
tstVisible	Only visible nodes are rendered.

Group

Types(see page 580)

File

VirtualTrees

10.4.148 TVSTTextType Enumeration

Not documented.

Pascal

```
TVSTTextType = (ttNormal, ttStatic);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
ttNormal	normal label of the node, this is also the text which can be edited
ttStatic	static (non-editable) text after the normal text

Group

Types(see page 580)

File

VirtualTrees

10.4.149 TVTAnimationOption Enumeration

Not documented.

Pascal

```
TVTAnimationOption = (toAnimatedToggle);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
toAnimatedToggle	Expanding and collapsing a node is animated (quick window scroll).

Group

Types(see page 580)

File

VirtualTrees

10.4.150 TVTAutoOption Enumeration

Not documented.

Pascal

```
TVTAutoOption = (toAutoDropExpand, toAutoExpand, toAutoScroll, toAutoScrollOnExpand, toAutoSort,
toAutoSpanColumns, toAutoTristateTracking, toAutoHideButtons, toAutoDeleteMovedNodes,
toDisableAutoscrollOnFocus, toAutoChangeScale, toAutoFreeOnCollapse, toDisableAutoscrollOnEdit);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
toAutoDropExpand	Expand node if it is the drop target for more than certain time.
toAutoExpand	Nodes are expanded (collapsed) when getting (losing) the focus.
toAutoScroll	Scroll if mouse is near the border while dragging or selecting.

toAutoScrollOnExpand	Scroll as many child nodes in view as possible after expanding a node.
toAutoSort	Sort tree when Header.SortColumn or Header.SortDirection change or sort node if child nodes are added.
toAutoSpanColumns	Large entries continue into next column(s) if there's no text in them (no clipping).
toAutoTristateTracking	Checkstates are automatically propagated for tri state check boxes.
toAutoHideButtons	Node buttons are hidden when there are child nodes, but all are invisible.
toAutoDeleteMovedNodes	Delete nodes which where moved in a drag operation (if not directed otherwise).
toDisableAutoscrollOnFocus	Disable scrolling a column entirely into view if it gets focused.
toAutoChangeScale	Change default node height automatically if the system's font scale is set to big fonts.
toAutoFreeOnCollapse	Frees any child node after a node has been collapsed (HasChildren flag stays there).

Group

Types(see page 580)

File

VirtualTrees

10.4.151 TVTButtonFillMode Enumeration

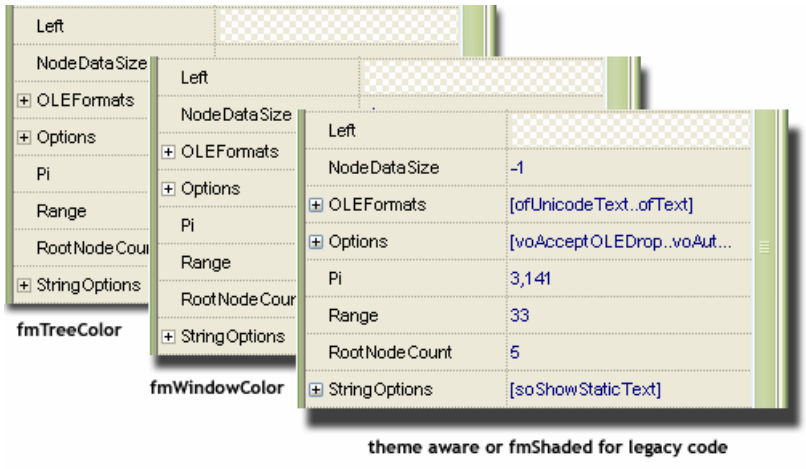
Determines how the interior of nodes buttons should be drawn.

Pascal

```
TVTButtonFillMode = (fmTreeColor, fmWindowColor, fmShaded, fmTransparent);
```

Description

Usually the little plus and minus buttons have just the color of the treeview but sometimes it looks better to use another kind of painting. This is particularly important when simulating Windows XP buttons on non-XP systems. The image below shows how the various modes look like:



Members

Members	Description
fmTreeColor	solid color, uses the tree's background color
fmWindowColor	solid color, uses clWindow
fmShaded	color gradient, Windows XP style (legacy code, use toThemeAware on Windows XP instead)
fmTransparent	transparent color, use the item's background color

Group

Types(see page 580)

File

VirtualTrees

10.4.152 TVTButtonStyle Enumeration

Not documented.

Pascal

```
TVTButtonStyle = (bsRectangle, bsTriangle);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
bsRectangle	traditional Windows look (plus/minus buttons)
bsTriangle	traditional Macintosh look

Group

Types(see page 580)

File

VirtualTrees

10.4.153 TVTColumnOption Enumeration

Not documented.

Pascal

```
TVTColumnOption = (coAllowClick, coDraggable, coEnabled, coParentBidiMode, coParentColor, coResizable,
coShowDropMark, coVisible, coAutoSpring, coFixed);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.154 TVTDragImageKind Enumeration

Not documented.

Pascal

```
TVTDragImageKind = (diComplete, diMainColumnOnly, diNoImage);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
diComplete	show a complete drag image with all columns, only visible columns are shown
diMainColumnOnly	show only the main column (the tree column)
diNoImage	don't show a drag image at all

Group

Types(see page 580)

File

VirtualTrees

10.4.155 TVTDragMoveRestriction Enumeration

Not documented.

Pascal

```
TVTDragMoveRestriction = (dmrNone, dmrHorizontalOnly, dmrVerticalOnly);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.156 TVTDragType Enumeration

Not documented.

Pascal

```
TVTDragType = (dtOLE, dtVCL);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.157 TVTDrawSelectionMode Enumeration

Not documented.

Pascal

```
TVTDrawSelectionMode = (smDottedRectangle, smBlendedRectangle);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
smDottedRectangle	same as DrawFocusRect
smBlendedRectangle	alpha blending, uses special colors (see TVTColors(see TVTColors Class, page 515))

Group

Types(see page 580)

File

VirtualTrees

10.4.158 TVTDropMarkMode Enumeration

Not documented.

Pascal

```
TVTDropMarkMode = (dmmNone, dmmLeft, dmmRight);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.159 TVTHeaderColumnLayout Enumeration

Not documented.

Pascal

```
TVTHeaderColumnLayout = (blGlyphLeft, blGlyphRight, blGlyphTop, blGlyphBottom);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.160 TVTHeaderOption Enumeration

Not documented.

Pascal

```
TVTHeaderOption = (hoAutoResize, hoColumnResize, hoDblClickResize, hoDrag, hoHotTrack, hoOwnerDraw,
hoRestrictDrag, hoShowHint, hoShowImages, hoShowSortGlyphs, hoVisible, hoAutoSpring);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
hoAutoResize	adjust a column so that the header never exceeds client width of owner control
hoColumnResize	resizing columns is allowed
hoDblClickResize	allows a column to resize itself to its largest entry
hoDrag	dragging columns is allowed
hoHotTrack	header captions are highlighted when mouse is over a particular column
hoOwnerDraw	header items with the owner draw style can be drawn by the application via event
hoRestrictDrag	header can only be dragged horizontally
hoShowHint	show application defined header hint
hoShowImages	show images
hoShowSortGlyphs	allow visible sort glyphs
hoVisible	header is visible

Group

Types(see page 580)

File

VirtualTrees

10.4.161 TVTHeaderPopupOption Enumeration

Not documented.

Pascal

```
TVTHeaderPopupOption = (poOriginalOrder, poAllowHideAll);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
poOriginalOrder	Show menu items in original column order as they were added to the tree.
poAllowHideAll	Allows to hide all columns, including the last one.

Group

Types(see page 580)

File

VTHeaderPopup

10.4.162 TVTMenuItem Type

Not documented.

Pascal

```
TVTMenuItem = TMenuItem;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VTHeaderPopup

10.4.163 TVTHeaderStyle Enumeration

Not documented.

Pascal

```
TVTHeaderStyle = (hsThickButtons, hsFlatButtons, hsPlates, hsXPStyle);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
hsThickButtons	TButton look and feel
hsFlatButtons	flatter look than hsThickButton, like an always raised flat TToolButton
hsPlates	flat TToolButton look and feel (raise on hover etc.)
hsXPStyle	Windows XP style

Group

Types(see page 580)

File

VirtualTrees

10.4.164 TVTHintMode Enumeration

Not documented.

Pascal

```
TVTHintMode = (hmDefault, hmHint, hmHintAndDefault, hmTooltip);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
hmDefault	show the hint of the control
hmHint	show node specific hint string returned by the application
hmHintAndDefault	same as hmHint but show the control's hint if no node is concerned
hmTooltip	show the text of the node if it isn't already fully shown

Group

Types(see page 580)

File

VirtualTrees

10.4.165 TVTImageInfoIndex Enumeration

Not documented.

Pascal

```
TVTImageInfoIndex = (iiNormal, iiState, iiCheck, iiOverlay);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.166 TVTImageKind Enumeration

Not documented.

Pascal

```
TVTImageKind = (ikNormal, ikSelected, ikState, ikOverlay);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.167 TVTIncrementalSearch Enumeration

Not documented.

Pascal

```
TVTIncrementalSearch = (isAll, isNone, isInitializedOnly, isVisibleOnly);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
isAll	search every node in tree, initialize if necessary
isNone	disable incremental search
isInitializedOnly	search only initialized nodes, skip others
isVisibleOnly	search only visible nodes, initialize if necessary

Group

Types(see page 580)

File

VirtualTrees

10.4.168 TVTInternalPaintOption Enumeration

Not documented.

Pascal

```
TVTInternalPaintOption = (poBackground, poColumnColor, poDrawFocusRect, poDrawSelection,
poDrawDropMark, poGridLines, poMainOnly, poSelectedOnly);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
poBackground	draw background image if there is any and it is enabled
poColumnColor	erase node's background with the column's color
poDrawFocusRect	draw focus rectangle around the focused node
poDrawSelection	draw selected nodes with the normal selection color
poDrawDropMark	draw drop mark if a node is currently the drop target

poGridLines	draw grid lines if enabled
poMainOnly	draw only the main column
poSelectedOnly	draw only selected nodes

Group

Types(see page 580)

File

VirtualTrees

10.4.169 TVTLineMode Enumeration

Not documented.

Pascal

```
TVTLineMode = (lmNormal, lmBands);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
lmNormal	usual tree lines (as in TTreeview)
lmBands	looks similar to a Nassi-Schneidermann diagram

Group

Types(see page 580)

File

VirtualTrees

10.4.170 TVTLineStyle Enumeration

Not documented.

Pascal

```
TVTLineStyle = (lsCustomStyle, lsDotted, lsSolid);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
lsCustomStyle	application provides a line pattern
lsDotted	usual dotted lines (default)
lsSolid	simple solid lines

Group

Types(see page 580)

File

VirtualTrees

10.4.171 TVTLineType Enumeration

Not documented.

Pascal

```
TVTLineType = (ltNone, ltBottomRight, ltTopDown, ltTopDownRight, ltRight, ltTopRight, ltLeft,
ltLeftBottom);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
ltNone	no line at all
ltBottomRight	a line from bottom to the center and from there to the right
ltTopDown	a line from top to bottom
ltTopDownRight	a line from top to bottom and from center to the right
ltRight	a line from center to the right
ltTopRight	a line from bottom to center and from there to the right special styles for alternative drawings of tree lines
ltLeft	a line from top to bottom at the left
ltLeftBottom	a combination of ltLeft and a line at the bottom from left to right

Group

Types(see page 580)

File

VirtualTrees

10.4.172 TVTMiscOption Enumeration

Not documented.

Pascal

```
TVTMiscOption = (toAcceptOLEDrop, toCheckSupport, toEditable, toFullRepaintOnResize, toGridExtensions,
toInitOnSave, toReportMode, toToggleOnDblClick, toWheelPanning, toReadOnly, toVariableNodeHeight,
toFullRowDrag);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
toAcceptOLEDrop	Register tree as OLE accepting drop target
toCheckSupport	Show checkboxes/radio buttons.
toEditable	Node captions can be edited.
toFullRepaintOnResize	Fully invalidate the tree when its window is resized (CS_HREDRAW/CS_VREDRAW).

toGridExtensions	Use some special enhancements to simulate and support grid behavior.
toInitOnSave	Initialize nodes when saving a tree to a stream.
toReportMode	Tree behaves like TListView in report mode.
toToggleOnDblClick	Toggle node expansion state when it is double clicked.
toWheelPanning	This option and toMiddleClickSelect are mutal exclusive, where panning has precedence.
toReadOnly	No action is executed and node editing is not possible.

Group

Types(see page 580)

File

VirtualTrees

10.4.173 TVTNodeAlignment Enumeration

Not documented.

Pascal

```
TVTNodeAlignment = (naFromBottom, naFromTop, naProportional);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
naFromBottom	the align member specifies amount of units (usually pixels) from top border of the node
naFromTop	align is to be measured from bottom
naProportional	align is to be measure in percent of the entire node height and relative to top

Group

Types(see page 580)

File

VirtualTrees

10.4.174 TVTNodeAttachMode Enumeration

Not documented.

Pascal

```
TVTNodeAttachMode = (amNoWhere, amInsertBefore, amInsertAfter, amAddChildFirst, amAddChildLast);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
amNoWhere	just for simplified tests, means to ignore the Add/Insert command
amInsertBefore	insert node just before destination (as sibling of destination)

amInsertAfter	insert node just after destionation (as sibling of destination)
amAddChildFirst	add node as first child of destination
amAddChildLast	add node as last child of destination

Group

Types(see page 580)

File

VirtualTrees

10.4.175 TVTScrollbarShowEvent Type

Not documented.

Pascal

```
TVTScrollbarShowEvent = procedure (Sender: TBaseVirtualTree; Bar: Integer; Show: Boolean) of object;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.176 TVTPaintOption Enumeration

Not documented.

Pascal

```
TVTPaintOption = (toHideFocusRect, toHideSelection, toHotTrack, toPopupMode, toShowBackground,
toShowButtons, toShowDropmark, toShowHorzGridLines, toShowRoot, toShowTreeLines, toShowVertGridLines,
toThemeAware, toUseBlendedImages, toGhostedIfUnfocused, toFullVertGridLines, toAlwaysHideSelection,
toUseBlendedSelection, toStaticBackground);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
toHideFocusRect	Avoid drawing the dotted rectangle around the currently focused node.
toHideSelection	Selected nodes are drawn as unselected nodes if the tree is unfocused.
toHotTrack	Track which node is under the mouse cursor.
toPopupMode	Paint tree as would it always have the focus (useful for tree combo boxes etc.)
toShowBackground	Use the background image if there's one.
toShowButtons	Display collapse/expand buttons left to a node.
toShowDropmark	Show the dropmark during drag'n drop operations.
toShowHorzGridLines	Display horizontal lines to simulate a grid.
toShowRoot	Show lines also at top level (does not show the hidden/internal root node).

toShowTreeLines	Display tree lines to show hierarchy of nodes.
toShowVertGridLines	Display vertical lines (depending on columns) to simulate a grid.
toThemeAware	Draw UI elements (header, tree buttons etc.) according to the current theme if enabled (Windows XP+ only, application must be themed).
toUseBlendedImages	Enable alpha blending for ghosted nodes or those which are being cut/copied.
toGhostedIfUnfocused	Ghosted images are still shown as ghosted if unfocused (otherwise the become non-ghosted images).
toFullVertGridLines	This option only has an effect if toShowVertGridLines is enabled too.
toAlwaysHideSelection	Do not draw node selection, regardless of focused state.
toUseBlendedSelection	Enable alpha blending for node selections.

Group

Types(see page 580)

File

VirtualTrees

10.4.177 TVTSearchDirection Enumeration

Not documented.

Pascal

```
TVTSearchDirection = (sdForward, sdBackward);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types(see page 580)

File

VirtualTrees

10.4.178 TVTSearchStart Enumeration

Not documented.

Pascal

```
TVTSearchStart = (ssAlwaysStartOver, ssLastHit, ssFocusedNode);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
ssAlwaysStartOver	always use the first/last node (depending on direction) to search from
ssLastHit	use the last found node
ssFocusedNode	use the currently focused node

Group

Types(see page 580)

File

VirtualTrees

10.4.179 TVTSelectionOption Enumeration

Not documented.

Pascal

```
TVTSelectionOption = (toDisableDrawSelection, toExtendedFocus, toFullRowSelect,
toLevelSelectConstraint, toMiddleClickSelect, toMultiSelect, toRightClickSelect,
toSiblingSelectConstraint, toCenterScrollIntoView, toSimpleDrawSelection);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
toDisableDrawSelection	Prevent user from selecting with the selection rectangle in multiselect mode.
toExtendedFocus	Entries other than in the main column can be selected, edited etc.
toFullRowSelect	Hit test as well as selection highlight are not constrained to the text of a node.
toLevelSelectConstraint	Constrain selection to the same level as the selection anchor.
toMiddleClickSelect	with the middle mouse button. This and toWheelPanning are mutual exclusive.
toMultiSelect	Allow more than one node to be selected.
toRightClickSelect	with the right mouse button.
toSiblingSelectConstraint	constrain selection to nodes with same parent
toCenterScrollIntoView	Center nodes vertically in the client area when scrolling into view.
toSimpleDrawSelection	Simplifies draw selection, so a node's caption does not need to intersect with the selection rectangle.

Group

Types(see page 580)

File

VirtualTrees

10.4.180 TVTStringOption Enumeration

Not documented.

Pascal

```
TVTStringOption = (toSaveCaptions, toShowStaticText, toAutoAcceptEditChange);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
toSaveCaptions	If set then the caption is automatically saved with the tree node, regardless of what is saved in the user data.

toShowStaticText	Show static text in a caption which can be differently formatted than the caption but cannot be edited.
toAutoAcceptEditChange	If not set then changes are cancelled.

Group

Types(see page 580)

File

VirtualTrees

10.4.181 TVTUpdateState Enumeration

Not documented.

Pascal

```
TVTUpdateState = (usBegin, usBeginSynch, usSynch, usUpdate, usEnd, usEndSynch);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

Members	Description
usBegin	The tree just entered the update state (BeginUpdate call for the first time).
usBeginSynch	The tree just entered the synch update state (BeginSynch call for the first time).
usSynch	Begin/EndSynch has been called but the tree did not change the update state.
usUpdate	Begin/EndUpdate has been called but the tree did not change the update state.
usEnd	The tree just left the update state (EndUpdate called for the last level).
usEndSynch	The tree just left the synch update state (EndSynch called for the last level).

Group

Types(see page 580)

File

VirtualTrees

10.5 Variables

These are all variables that are contained in this documentation.

Group

Virtual Treeview(see page 71)

Variables

CF_CSV(see CF_CSV Variable, page 658)
Not documented.

CF_HTML(see CF_HTML Variable, page 658)
Not documented.

CF_VIRTUALTREE(see CF_VIRTUALTREE Variable, page 658)

Not documented.

CF_VRTF(see CF_VRTF Variable, page 659)
Not documented.

CF_VRTFNOOBJS(see CF_VRTFNOOBJS Variable, page 659)
Not documented.

CF_VTREFERENCE(see CF_VTREFERENCE Variable, page 659)
Not documented.

ClipboardDescriptions(see ClipboardDescriptions Variable, page 660)
Not documented.

DarkCheckImages(see DarkCheckImages Variable, page 660)
Not documented.

DarkTickImages(see DarkTickImages Variable, page 660)
Not documented.

FlatImages(see FlatImages Variable, page 661)
Not documented.

HintFont(see HintFont Variable, page 661)
Not documented.

HintWindowDestroyed(see HintWindowDestroyed Variable, page 661)
Not documented.

Initialized(see Initialized Variable, page 662)
Not documented.

InternalClipboardFormats(see InternalClipboardFormats Variable, page 662)
Not documented.

IsWin2K(see IsWin2K Variable, page 662)
Not documented.

IsWinNT(see IsWinNT Variable, page 663)
Not documented.

IsWinXP(see IsWinXP Variable, page 663)
Not documented.

LightCheckImages(see LightCheckImages Variable, page 663)
Not documented.

LightTickImages(see LightTickImages Variable, page 664)
Not documented.

MMXAvailable(see MMXAvailable Variable, page 664)
Not documented.

NeedToUnitialize(see NeedToUnitialize Variable, page 664)
Not documented.

StandardOLEFormat(see StandardOLEFormat Variable, page 665)
Not documented.

SystemCheckImages(see SystemCheckImages Variable, page 665)
Not documented.

SystemFlatCheckImages(see SystemFlatCheckImages Variable, page 665)
Not documented.

UtilityImages(see UtilityImages Variable, page 666)
Not documented.

Watcher(see Watcher Variable, page 666)
Not documented.

WorkerThread(see WorkerThread Variable, page 666)
Not documented.

WorkEvent(see WorkEvent Variable, page 667)
Not documented.

XPImages(see XPImages Variable, page 667)
Not documented.

Legend

Variable

10.5.1 CF_CSV Variable

Not documented.

Pascal

```
CF_CSV: Word;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.2 CF_HTML Variable

Not documented.

Pascal

```
CF_HTML: Word;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.3 CF_VIRTUALTREE Variable

Not documented.

Pascal

```
CF_VIRTUALTREE: Word;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.4 CF_VRTF Variable

Not documented.

Pascal

```
CF_VRTF: Word;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.5 CF_VRTFNOOBJS Variable

Not documented.

Pascal

```
CF_VRTFNOOBJS: Word;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.6 CF_VTREFERENCE Variable

Not documented.

Pascal

```
CF_VTREFERENCE: Word;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.7 ClipboardDescriptions Variable

Not documented.

Pascal

```
ClipboardDescriptions: array [1..CF_MAX - 1] of TClipboardFormatEntry = ( (ID: CF_TEXT; Description:
'Plain text'), (ID: CF_BITMAP; Description: 'Windows bitmap'), (ID: CF_METAFILEPICT; Description:
'Windows metafile'), (ID: CF_SYLK; Description: 'Symbolic link'), (ID: CF_DIF; Description: 'Data
interchange format'), (ID: CF_TIFF; Description: 'Tiff image'), (ID: CF_OEMTEXT; Description: 'OEM
text'), (ID: CF_DIB; Description: 'DIB image'), (ID: CF_PALETTE; Description: 'Palette data'), (ID:
CF_PENDATA; Description: 'Pen data'), (ID: CF_RIFF; Description: 'Riff audio data'), (ID: CF_WAVE;
Description: 'Wav audio data'), (ID: CF_UNICODETEXT; Description: 'Unicode text'), (ID:
CF_ENHMETAFILE; Description: 'Enhanced metafile image'), (ID: CF_HDROP; Description: 'File name(s)'),
(ID: CF_LOCALE; Description: 'Locale descriptor') );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.8 DarkCheckImages Variable

Not documented.

Pascal

```
DarkCheckImages: TImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.9 DarkTickImages Variable

Not documented.

Pascal

```
DarkTickImages: TImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.10 FlatImages Variable

Not documented.

Pascal

```
FlatImages: TImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.11 HintFont Variable

Not documented.

Pascal

```
HintFont: TFont;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.12 HintWindowDestroyed Variable

Not documented.

Pascal

```
HintWindowDestroyed: Boolean = True;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.13 Initialized Variable

Not documented.

Pascal

```
Initialized: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.14 InternalClipboardFormats Variable

Not documented.

Pascal

```
InternalClipboardFormats: TClipboardFormatList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.15 IsWin2K Variable

Not documented.

Pascal

```
IsWin2K: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.16 IsWinNT Variable

Not documented.

Pascal

```
IsWinNT: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.17 IsWinXP Variable

Not documented.

Pascal

```
IsWinXP: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.18 LightCheckImages Variable

Not documented.

Pascal

```
LightCheckImages: TImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.19 LightTickImages Variable

Not documented.

Pascal

```
LightTickImages: TImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.20 MMXAvailable Variable

Not documented.

Pascal

```
MMXAvailable: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.21 NeedToUnitialize Variable

Not documented.

Pascal

```
NeedToUnitialize: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.22 StandardOLEFormat Variable

Not documented.

Pascal

```
StandardOLEFormat: TFormatEtc = ( cfFormat: 0; ptd: nil; dwAspect: DVASPECT_CONTENT; lindex: -1;
tymed: TYMED_ISTREAM or TYMED_HGLOBAL; );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.23 SystemCheckImages Variable

Not documented.

Pascal

```
SystemCheckImages: TImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.24 SystemFlatCheckImages Variable

Not documented.

Pascal

```
SystemFlatCheckImages: TImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.25 UtilityImages Variable

Not documented.

Pascal

```
UtilityImages: TImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.26 Watcher Variable

Not documented.

Pascal

```
Watcher: TCriticalSection;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.27 WorkerThread Variable

Not documented.

Pascal

```
WorkerThread: TWorkerThread;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.28 WorkEvent Variable

Not documented.

Pascal

```
WorkEvent: THandle;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.5.29 XPImages Variable

Not documented.

Pascal

```
XPImages: TImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables(see page 656)

File

VirtualTrees

10.6 Constants

These are all constants that are contained in this documentation.

Constants

AlignmentToDrawFlag(see AlignmentToDrawFlag Constant, page 672)
Not documented.

AllocIncrement(see AllocIncrement Constant, page 672)
Not documented.

BaseChunk(see BaseChunk Constant, page 672)
Not documented.

CacheThreshold(see CacheThreshold Constant, page 673)
Number of nodes a tree must at least have to start caching and at the same time the maximum number of nodes between two cache entries.

CaptionChunk(see CaptionChunk Constant, page 673)
Not documented.

CFSTR_CSV(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

CFSTR_HTML(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

CFSTR_RTF(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

CFSTR_RTFNOOBJS(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

CFSTR_VIRTUALTREE(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

CFSTR_VTREFERENCE(see CFSTR_CSV Constant, page 673)
Contains the registration string for certain clipboard formats.

ChangeTimer(see ChangeTimer Constant, page 674)
Not documented.

ckButtonDisabled(see Check button image indices, page 674)
ckButtonHot(see Check button image indices, page 674)
ckButtonNormal(see Check button image indices, page 674)
ckButtonPressed(see Check button image indices, page 674)
ckCheckCheckedDisabled(see Check button image indices, page 674)
ckCheckCheckedHot(see Check button image indices, page 674)
ckCheckCheckedNormal(see Check button image indices, page 674)
ckCheckCheckedPressed(see Check button image indices, page 674)
ckCheckMixedDisabled(see Check button image indices, page 674)
ckCheckMixedHot(see Check button image indices, page 674)
ckCheckMixedNormal(see Check button image indices, page 674)
ckCheckMixedPressed(see Check button image indices, page 674)
ckCheckUncheckedDisabled(see Check button image indices, page 674)
ckCheckUncheckedHot(see Check button image indices, page 674)
ckCheckUncheckedNormal(see Check button image indices, page 674)
ckCheckUncheckedPressed(see Check button image indices, page 674)
ckEmpty(see Check button image indices, page 674)
ckRadioCheckedDisabled(see Check button image indices, page 674)
ckRadioCheckedHot(see Check button image indices, page 674)
ckRadioCheckedNormal(see Check button image indices, page 674)
ckRadioCheckedPressed(see Check button image indices, page 674)
ckRadioUncheckedDisabled(see Check button image indices, page 674)
ckRadioUncheckedHot(see Check button image indices, page 674)
ckRadioUncheckedNormal(see Check button image indices, page 674)
ckRadioUncheckedPressed(see Check button image indices, page 674)
ClipboardStates(see ClipboardStates Constant, page 675)
Not documented.

CLSID_DragDropHelper(see CLSID_DragDropHelper Constant, page 675)
Not documented.

CM_AUTOADJUST(see CM_AUTOADJUST Constant, page 675)
Not documented.

CM_DENYSUBCLASSING(see CM_DENYSUBCLASSING Constant, page 676)
Not documented.

Copyright(see Copyright Constant, page 676)
Not documented.

crHeaderSplit(see crHeaderSplit Constant, page 676)
Not documented.

DefaultAnimationOptions(see DefaultAnimationOptions Constant, page 677)
Not documented.

DefaultAutoOptions(see DefaultAutoOptions Constant, page 677)
Not documented.

DefaultColumnOptions(see DefaultColumnOptions Constant, page 677)
Not documented.

DefaultMiscOptions(see DefaultMiscOptions Constant, page 678)
Not documented.

DefaultPaintOptions(see DefaultPaintOptions Constant, page 678)
Not documented.

DefaultScrollUpdateFlags(see DefaultScrollUpdateFlags Constant, page 678)
Not documented.

DefaultSelectionOptions(see DefaultSelectionOptions Constant, page 679)
Not documented.

DefaultStringOptions(see DefaultStringOptions Constant, page 679)
Not documented.

EditTimer(see EditTimer Constant, page 679)
Not documented.

ExpandTimer(see ExpandTimer Constant, page 680)
Not documented.

FadeAnimationStepCount(see FadeAnimationStepCount Constant, page 680)
Not documented.

Grays(see Grays Constant, page 680)
Not documented.

hcTFCannotSetUserData(see hcTFCannotSetUserData Constant, page 681)
Not documented.

hcTFClipboardFailed(see hcTFClipboardFailed Constant, page 681)
Not documented.

hcTFCorruptStream1(see hcTFCorruptStream1 Constant, page 681)
Not documented.

hcTFCorruptStream2(see hcTFCorruptStream2 Constant, page 682)
Not documented.

hcTFEditLinkIsNil(see hcTFEditLinkIsNil Constant, page 682)
Not documented.

hcTFStreamTooSmall(see hcTFStreamTooSmall Constant, page 682)
Not documented.

hcTFWrongMoveError(see hcTFWrongMoveError Constant, page 683)
Not documented.

hcTFWrongStreamFormat(see hcTFWrongStreamFormat Constant, page 683)
Not documented.

hcTFWrongStreamVersion(see hcTFWrongStreamVersion Constant, page 683)

Not documented.

HeaderTimer(see HeaderTimer Constant, page 684)
Not documented.

IID_IDragSourceHelper(see IID_IDragSourceHelper Constant, page 684)
Not documented.

IID_IDropTarget(see IID_IDropTarget Constant, page 684)
Not documented.

IID_IDropTargetHelper(see IID_IDropTargetHelper Constant, page 685)
Not documented.

InvalidColumn(see InvalidColumn Constant, page 685)
Not documented.

MagicID(see MagicID Constant, page 685)
Not documented.

MinimumTimerInterval(see MinimumTimerInterval Constant, page 686)
Not documented.

MouseButtonDown(see MouseButtonDown Constant, page 686)
Not documented.

NoColumn(see NoColumn Constant, page 686)
Not documented.

NodeChunk(see NodeChunk Constant, page 687)
Not documented.

OptionMap(see OptionMap Constant, page 687)
Not documented.

PressedState(see PressedState Constant, page 687)
Not documented.

RTLFlag(see RTLFlag Constant, page 688)
Not documented.

SCannotSetUserData(see SCannotSetUserData Constant, page 688)
Not documented.

SClipboardFailed(see SClipboardFailed Constant, page 688)
Not documented.

SCorruptStream1(see SCorruptStream1 Constant, page 689)
Not documented.

SCorruptStream2(see SCorruptStream2 Constant, page 689)
Not documented.

ScrollTimer(see ScrollTimer Constant, page 689)
Not documented.

SearchTimer(see SearchTimer Constant, page 690)
Not documented.

SEditLinkIsNil(see SEditLinkIsNil Constant, page 690)
Not documented.

ShadowSize(see ShadowSize Constant, page 690)
Size in pixels of the hint shadow.

SID_IDragSourceHelper(see SID_IDragSourceHelper Constant, page 691)
Not documented.

SID_IDropTarget(see SID_IDropTarget Constant, page 691)
Not documented.

SID_IDropTargetHelper(see SID_IDropTargetHelper Constant, page 691)
Not documented.

SStreamTooSmall(see SStreamTooSmall Constant, page 692)
Not documented.

StructureChangeTimer(see StructureChangeTimer Constant, page 692)
Not documented.

SWrongMoveError(see SWrongMoveError Constant, page 692)
Not documented.

SWrongStreamFormat(see SWrongStreamFormat Constant, page 693)
Not documented.

SWrongStreamVersion(see SWrongStreamVersion Constant, page 693)
Not documented.

SysGrays(see SysGrays Constant, page 693)
Not documented.

TreeNodeSize(see TreeNodeSize Constant, page 694)
Not documented.

UnpressedState(see UnpressedState Constant, page 694)
Not documented.

UserChunk(see UserChunk Constant, page 694)
Not documented.

UtilityImageSize(see UtilityImageSize Constant, page 695)
Not documented.

VTHeaderStreamVersion(see VTHeaderStreamVersion Constant, page 695)
Not documented.

VTTreeStreamVersion(see VTTreeStreamVersion Constant, page 695)
Not documented.

VTVersion(see VTVersion Constant, page 696)
Not documented.

WideCR(see WideCR Constant, page 696)
Not documented.

WideLF(see WideLF Constant, page 696)
Not documented.

WideLineSeparator(see WideLineSeparator Constant, page 697)
Not documented.

WideNull(see WideNull Constant, page 697)
Not documented.

WM_CHANGESTATE(see WM_CHANGESTATE Constant, page 697)
Not documented.

XPDarkGradientColor(see XPDarkGradientColor Constant, page 698)
Not documented.

XPDarkSplitBarColor(see XPDarkSplitBarColor Constant, page 698)
Not documented.

XPDownInnerLineColor(see XPDownInnerLineColor Constant, page 698)
Not documented.

XPDownMiddleLineColor(see XPDownMiddleLineColor Constant, page 699)
Not documented.

XPDownOuterLineColor(see XPDownOuterLineColor Constant, page 699)
Not documented.

XPLightSplitBarColor(see XPLightSplitBarColor Constant, page 699)
Not documented.

XPMainHeaderColorDown(see XPMainHeaderColorDown Constant, page 700)
Not documented.

XPMainHeaderColorHover(see XPMainHeaderColorHover Constant, page 700)
Not documented.

XPMainHeaderColorUp(see XPMainHeaderColorUp Constant, page 700)

Not documented.

Group

Virtual Treeview(see page 71)

Topics

Check button image indices(see page 674)

Legend

Constant

10.6.1 AlignmentToDrawFlag Constant

Not documented.

Pascal

```
AlignmentToDrawFlag: array[TAlignment] of Cardinal = (DT_LEFT, DT_RIGHT, DT_CENTER);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.2 AllocIncrement Constant

Not documented.

Pascal

```
AllocIncrement = 4096;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.3 BaseChunk Constant

Not documented.

Pascal

```
BaseChunk = 2;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.4 CacheThreshold Constant

Number of nodes a tree must at least have to start caching and at the same time the maximum number of nodes between two cache entries.

Pascal

```
CacheThreshold = 2000;
```

Description

Number of nodes a tree must at least have to start caching and at the same time the maximum number of nodes between two cache entries.

Group

Constants(see page 667)

File

VirtualTrees

10.6.5 CaptionChunk Constant

Not documented.

Pascal

```
CaptionChunk = 3;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.6 CFSTR_CSV Constant

Contains the registration string for certain clipboard formats.

Pascal

```
CFSTR_VIRTUALTREE = 'Virtual Tree Data';
CFSTR_VTREFERENCE = 'Virtual Tree Reference';
CFSTR_HTML = 'HTML Format';
CFSTR_RTF = 'Rich Text Format';
CFSTR_RTFNOOBJS = 'Rich Text Format Without Objects';
CFSTR_CSV = 'CSV';
```

Description

Some of the clipboard formats in the system, like CF_HDROP, are registered by Windows itself. For rich text, html, csv and other data first the formats must be registered with the clipboard. The identifier returned by the registration code is used to unregister the format later and to identify the format when transferring data or enumerating the clipboard formats. The following formats are registered by Virtual Treeview:

- CVS: comma separated values, a tabular data format.
- HTML: text data with text formatting and structured like a big table. Unicode is supported as well (UTF-8).
- RTF: rich text format, similar to HTML, but more complex and also a bit older.
- RTFNOOBJS: like RTF but without embedded objects (not used by Virtual Treeview).
- VIRTUALTREEVIEW: serialized treeview data. This is the native tree format and the only one directly accepted by the control.
- VTREFERENCE: a special format to pass on a reference of the sender treeview. If both, sender and receiver, live in the same process this reference can be used to directly access the sender treeview, without COM interecption.

Group

Constants(see page 667)

File

VirtualTrees

10.6.7 ChangeTimer Constant

Not documented.

Pascal

```
ChangeTimer = 5;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.8 Check button image indices

Constants(see page 667)

10.6.9 ClipboardStates Constant

Not documented.

Pascal

```
ClipboardStates = [tsCopyPending, tsCutPending];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.10 CLSID_DragDropHelper Constant

Not documented.

Pascal

```
CLSID_DragDropHelper: TGUID = (D1: $4657278A; D2: $411B; D3: $11D2; D4: ($83, $9A, $00, $C0, $4F, $D9,
$18, $D0));
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.11 CM_AUTOADJUST Constant

Not documented.

Pascal

```
CM_AUTOADJUST = CM_BASE + 2005;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.12 CM_DENYSUBCLASSING Constant

Not documented.

Pascal

```
CM_DENYSUBCLASSING = CM_BASE + 2000;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.13 Copyright Constant

Not documented.

Pascal

```
Copyright: string = 'Virtual Treeview © 1999, 2003 Mike Lischke';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.14 crHeaderSplit Constant

Not documented.

Pascal

```
crHeaderSplit = TCursor(100);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.15 DefaultAnimationOptions Constant

Not documented.

Pascal

```
DefaultAnimationOptions = [];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.16 DefaultAutoOptions Constant

Not documented.

Pascal

```
DefaultAutoOptions = [toAutoDropExpand, toAutoTristateTracking, toAutoScrollOnExpand,
toAutoDeleteMovedNodes];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.17 DefaultColumnOptions Constant

Not documented.

Pascal

```
DefaultColumnOptions = [coAllowClick, coDraggable, coEnabled, coParentColor, coParentBidiMode,
coResizable, coShowDropmark, coVisible];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.18 DefaultMiscOptions Constant

Not documented.

Pascal

```
DefaultMiscOptions = [toAcceptOLEDrop, toFullRepaintOnResize, toInitOnSave, toToggleOnDblClick,
toWheelPanning];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.19 DefaultPaintOptions Constant

Not documented.

Pascal

```
DefaultPaintOptions = [toShowButtons, toShowButtons, toShowDropmark, toShowTreeLines, toShowRoot,
toThemeAware, toUseBlendedImages];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.20 DefaultScrollUpdateFlags Constant

Not documented.

Pascal

```
DefaultScrollUpdateFlags = [suoRepaintHeader, suoRepaintScrollbars, suoScrollClientArea,
suoUpdateNCArea];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.21 DefaultSelectionOptions Constant

Not documented.

Pascal

```
DefaultSelectionOptions = [];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.22 DefaultStringOptions Constant

Not documented.

Pascal

```
DefaultStringOptions = [toSaveCaptions, toAutoAcceptEditChange];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.23 EditTimer Constant

Not documented.

Pascal

```
EditTimer = 2;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.24 ExpandTimer Constant

Not documented.

Pascal

```
ExpandTimer = 1;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.25 FadeAnimationStepCount Constant

Not documented.

Pascal

```
FadeAnimationStepCount = 255;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.26 Grays Constant

Not documented.

Pascal

```
Grays: array[0..3] of TColor = (clWhite, clSilver, clGray, clBlack);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.27 hcTFCannotSetUserData Constant

Not documented.

Pascal

```
hcTFCannotSetUserData = 2008;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.28 hcTFClipboardFailed Constant

Not documented.

Pascal

```
hcTFClipboardFailed = 2007;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.29 hcTFCorruptStream1 Constant

Not documented.

Pascal

```
hcTFCorruptStream1 = 2005;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.30 hcTFCorruptStream2 Constant

Not documented.

Pascal

```
hcTFCorruptStream2 = 2006;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.31 hcTFEditLinkIsNil Constant

Not documented.

Pascal

```
hcTFEditLinkIsNil = 2000;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.32 hcTFStreamTooSmall Constant

Not documented.

Pascal

```
hcTFStreamTooSmall = 2004;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.33 hcTFWrongMoveError Constant

Not documented.

Pascal

```
hcTFWrongMoveError = 2001;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.34 hcTFWrongStreamFormat Constant

Not documented.

Pascal

```
hcTFWrongStreamFormat = 2002;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.35 hcTFWrongStreamVersion Constant

Not documented.

Pascal

```
hcTFWrongStreamVersion = 2003;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.36 HeaderTimer Constant

Not documented.

Pascal

```
HeaderTimer = 3;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.37 IID_IDragSourceHelper Constant

Not documented.

Pascal

```
IID_IDragSourceHelper: TGUID = (D1: $DE5BF786; D2: $477A; D3: $11D2; D4: ($83, $9D, $00, $C0, $4F,
$D9, $18, $D0));
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.38 IID_IDropTarget Constant

Not documented.

Pascal

```
IID_IDropTarget: TGUID = (D1: $00000122; D2: $0000; D3: $0000; D4: ($C0, $00, $00, $00, $00, $00, $00,
$46));
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.39 IID_IDropTargetHelper Constant

Not documented.

Pascal

```
IID_IDropTargetHelper: TGUID = (D1: $4657278B; D2: $411B; D3: $11D2; D4: ($83, $9A, $00, $C0, $4F,
$D9, $18, $D0));
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.40 InvalidColumn Constant

Not documented.

Pascal

```
InvalidColumn = -2;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.41 MagicID Constant

Not documented.

Pascal

```
MagicID: TMagicID = (#$2045, 'V', 'T', WideChar(VTTreeStreamVersion), ' ', #$2046);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.42 MinimumTimerInterval Constant

Not documented.

Pascal

```
MinimumTimerInterval = 1;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.43 MouseButtonDown Constant

Not documented.

Pascal

```
MouseButtonDown = [tsLeftButtonDown, tsMiddleButtonDown, tsRightButtonDown];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.44 NoColumn Constant

Not documented.

Pascal

```
NoColumn = -1;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.45 NodeChunk Constant

Not documented.

Pascal

```
NodeChunk = 1;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.46 OptionMap Constant

Not documented.

Pascal

```
OptionMap: array[TOldVTOption] of Integer = ( Ord(toAcceptOLEDrop), Ord(toAnimatedToggle),
Ord(toAutoDropExpand), Ord(toAutoExpand), Ord(toAutoScroll), Ord(toAutoSort), Ord(toAutoSpanColumns),
Ord(toAutoTristateTracking), Ord(toCheckSupport), Ord(toDisableDrawSelection), Ord(toEditable),
Ord(toExtendedFocus), Ord(toFullRowSelect), Ord(toGridExtensions), Ord(toHideFocusRect),
Ord(toHideSelection), Ord(toHotTrack), Ord(toInitOnSave), Ord(toLevelSelectConstraint),
Ord(toMiddleClickSelect), Ord(toMultiSelect), Ord(toRightClickSelect), Ord(toPopupMode),
Ord(toShowBackground), Ord(toShowButtons), Ord(toShowDropmark), Ord(toShowHorzGridLines),
Ord(toShowRoot), Ord(toShowTreeLines), Ord(toShowVertGridLines), Ord(toSiblingSelectConstraint),
Ord(toToggleOnDblClick) );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.47 PressedState Constant

Not documented.

Pascal

```
PressedState: array[TCheckState] of TCheckState = ( csUncheckedPressed, csUncheckedPressed,
csCheckedPressed, csCheckedPressed, csMixedPressed, csMixedPressed );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.48 RTLFlag Constant

Not documented.

Pascal

```
RTLFlag: array[Boolean] of Integer = (0, ETO_RTLREADING);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.49 SCannotSetUserData Constant

Not documented.

Pascal

```
SCannotSetUserData = 'Cannot set initial user data because there is not enough user data space
allocated.';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.50 SClipboardFailed Constant

Not documented.

Pascal

```
SClipboardFailed = 'Clipboard operation failed.';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.51 SCorruptStream1 Constant

Not documented.

Pascal

```
SCorruptStream1 = 'Stream data corrupt. A node''s anchor chunk is missing.';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.52 SCorruptStream2 Constant

Not documented.

Pascal

```
SCorruptStream2 = 'Stream data corrupt. Unexpected data after node''s end position.';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.53 ScrollTimer Constant

Not documented.

Pascal

```
ScrollTimer = 4;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.54 SearchTimer Constant

Not documented.

Pascal

```
SearchTimer = 7;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.55 SEditLinkIsNil Constant

Not documented.

Pascal

```
SEditLinkIsNil = 'Edit link must not be nil.';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.56 ShadowSize Constant

Size in pixels of the hint shadow.

Pascal

```
ShadowSize = 5;
```

Description

This value has no influence on Win2K and XP systems as those OSes have native shadow support. Set it to 0 if you don't want shadows on the other systems.

Group

Constants(see page 667)

File

VirtualTrees

10.6.57 SID_IDragSourceHelper Constant

Not documented.

Pascal

```
SID_IDragSourceHelper = '{DE5BF786-477A-11D2-839D-00C04FD918D0}';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.58 SID_IDropTarget Constant

Not documented.

Pascal

```
SID_IDropTarget = '{00000122-0000-0000-C000-000000000046}';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.59 SID_IDropTargetHelper Constant

Not documented.

Pascal

```
SID_IDropTargetHelper = '{4657278B-411B-11D2-839A-00C04FD918D0}';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.60 SStreamTooSmall Constant

Not documented.

Pascal

```
SStreamTooSmall = 'Unable to load tree structure, not enough data available.';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.61 StructureChangeTimer Constant

Not documented.

Pascal

```
StructureChangeTimer = 6;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.62 SWrongMoveError Constant

Not documented.

Pascal

```
SWrongMoveError = 'Target node cannot be a child node of the node to be moved.';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.63 SWrongStreamFormat Constant

Not documented.

Pascal

```
SWrongStreamFormat = 'Unable to load tree structure, the format is wrong.';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.64 SWrongStreamVersion Constant

Not documented.

Pascal

```
SWrongStreamVersion = 'Unable to load tree structure, the version is unknown.';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.65 SysGrays Constant

Not documented.

Pascal

```
SysGrays: array[0..3] of TColor = (clWindow, clBtnFace, clBtnShadow, clBtnText);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.66 TreeNodeSize Constant

Not documented.

Pascal

```
TreeNodeSize = (SizeOf(TVirtualNode) + 3) and not 3;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.67 UnpressedState Constant

Not documented.

Pascal

```
UnpressedState: array[TCheckState] of TCheckState = ( csUncheckedNormal, csUncheckedNormal,
csCheckedNormal, csCheckedNormal, csMixedNormal, csMixedNormal );
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.68 UserChunk Constant

Not documented.

Pascal

```
UserChunk = 4;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.69 UtilityImageSize Constant

Not documented.

Pascal

```
UtilityImageSize = 16;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.70 VTHeaderStreamVersion Constant

Not documented.

Pascal

```
VTHeaderStreamVersion = 3;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.71 VTTreeStreamVersion Constant

Not documented.

Pascal

```
VTTreeStreamVersion = 2;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.72 VTVersion Constant

Not documented.

Pascal

```
VTVersion = '4.4.2';
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.73 WideCR Constant

Not documented.

Pascal

```
WideCR = WideChar(#13);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.74 WideLF Constant

Not documented.

Pascal

```
WideLF = WideChar(#10);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.75 WideLineSeparator Constant

Not documented.

Pascal

```
WideLineSeparator = WideChar(#2028);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.76 WideNull Constant

Not documented.

Pascal

```
WideNull = WideChar(#0);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.77 WM_CHANGESTATE Constant

Not documented.

Pascal

```
WM_CHANGESTATE = WM_APP + 32;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.78 XPDarkGradientColor Constant

Not documented.

Pascal

```
XPDarkGradientColor = $B8C7CB;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.79 XPDarkSplitBarColor Constant

Not documented.

Pascal

```
XPDarkSplitBarColor = $B2C5C7;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.80 XPDownInnerLineColor Constant

Not documented.

Pascal

```
XPDownInnerLineColor = $C9D1D0;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.81 XPDownMiddleLineColor Constant

Not documented.

Pascal

```
XPDownMiddleLineColor = $B8C2C1;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.82 XPDownOuterLineColor Constant

Not documented.

Pascal

```
XPDownOuterLineColor = $97A5A5;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.83 XPLightSplitBarColor Constant

Not documented.

Pascal

```
XPLightSplitBarColor = $FFFFFF;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.84 XPMainHeaderColorDown Constant

Not documented.

Pascal

```
XPMainHeaderColorDown = $D8DFDE;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.85 XPMainHeaderColorHover Constant

Not documented.

Pascal

```
XPMainHeaderColorHover = $F3F8FA;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.6.86 XPMainHeaderColorUp Constant

Not documented.

Pascal

```
XPMainHeaderColorUp = $DBEAEB;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants(see page 667)

File

VirtualTrees

10.7 Symbol Reference

These are all symbols available in this documentation.

Group

Virtual Treeview(see page 71)

Interfaces

IDragSourceHelper(see IDragSourceHelper Interface, page 701)
Not documented.

IDropTargetHelper(see IDropTargetHelper Interface, page 702)
Not documented.

IVTDragManager(see IVTDragManager Interface, page 704)
Not documented.

IVTEditLink(see IVTEditLink Interface, page 707)
Interface which is used for communication between the treeview and a node editor.

Legend

Class

10.7.1 Interfaces

10.7.1.1 IDragSourceHelper Interface

Not documented.

Pascal

```
IDragSourceHelper = interface(IUnknown);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Symbol Reference(see page 701)

Methods

InitializeFromBitmap(see IDragSourceHelper.InitializeFromBitmap Method , page 702)
Not documented.

InitializeFromWindow(see IDragSourceHelper.InitializeFromWindow Method , page 702)

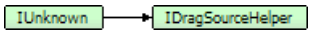
Not documented.

Legend

public

Method

Class Hierarchy



File

VirtualTrees

10.7.1.1.1 IDragSourceHelper.InitializeFromBitmap Method

Not documented.

Pascal

```
[SID_IDragSourceHelper]
function InitializeFromBitmap(var SHDragImage: TSHDragImage; pDataObject: IDataObject): HRESULT;
stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IDragSourceHelper Interface(see page 701)

10.7.1.1.2 IDragSourceHelper.InitializeFromWindow Method

Not documented.

Pascal

```
function InitializeFromWindow(Window: HWND; var ppt: TPoint; pDataObject: IDataObject): HRESULT;
stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IDragSourceHelper Interface(see page 701)

10.7.1.2 IDropTargetHelper Interface

Not documented.

Pascal

```
IDropTargetHelper = interface(IUnknown);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Symbol Reference(see page 701)

Methods

DragEnter(see IDropTargetHelper.DragEnter Method , page 703)
Not documented.

DragLeave(see IDropTargetHelper.DragLeave Method , page 703)
Not documented.

DragOver(see IDropTargetHelper.DragOver Method , page 704)
Not documented.

Drop(see IDropTargetHelper.Drop Method , page 704)
Not documented.

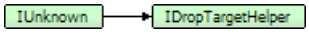
Show(see IDropTargetHelper.Show Method , page 704)
Not documented.

Legend

public

Method

Class Hierarchy



File

VirtualTrees

10.7.1.2.1 IDropTargetHelper.DragEnter Method

Not documented.

Pascal

```
[SID_IDropTargetHelper]
function DragEnter(hwndTarget: HWND; pDataObject: IDataObject; var ppt: TPoint; dwEffect: Integer):
HRESULT; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IDropTargetHelper Interface(see page 702)

10.7.1.2.2 IDropTargetHelper.DragLeave Method

Not documented.

Pascal

```
function DragLeave: HRESULT; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IDropTargetHelper Interface(see page 702)

10.7.1.2.3 IDropTargetHelper.DragOver Method

Not documented.

Pascal

```
function DragOver(var ppt: TPoint; dwEffect: Integer): HRESULT; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IDropTargetHelper Interface(see page 702)

10.7.1.2.4 IDropTargetHelper.Drop Method

Not documented.

Pascal

```
function Drop(pDataObject: IDataObject; var ppt: TPoint; dwEffect: Integer): HRESULT; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IDropTargetHelper Interface(see page 702)

10.7.1.2.5 IDropTargetHelper.Show Method

Not documented.

Pascal

```
function Show(fShow: Boolean): HRESULT; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IDropTargetHelper Interface(see page 702)

10.7.1.3 IVTDragManager Interface

Not documented.

Pascal

```
IVTDragManager = interface(IUnknown);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Symbol Reference(see page 701)

Methods

ForceDragLeave(see IVTDragManager.ForceDragLeave Method , page 706)

Not documented.

GetDataObject(see IVTDragManager.GetDataObject Method , page 706)

Not documented.

GetDragSource(see IVTDragManager.GetDragSource Method , page 707)

Not documented.

GetDropTargetHelperSupported(see IVTDragManager.GetDropTargetHelperSupported Method , page 707)

Not documented.

GetIsDropTarget(see IVTDragManager.GetIsDropTarget Method , page 707)

Not documented.

Properties

DataObject(see IVTDragManager.DataObject Property, page 705)

Not documented.

DragSource(see IVTDragManager.DragSource Property, page 705)

Not documented.

DropTargetHelperSupported(see IVTDragManager.DropTargetHelperSupported Property, page 706)

Not documented.

IsDropTarget(see IVTDragManager.IsDropTarget Property, page 706)

Not documented.

Legend

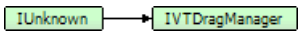
public

Property

read only

Method

Class Hierarchy



File

VirtualTrees

10.7.1.3.1 IVTDragManager.DataObject Property

Not documented.

Pascal

```
property DataObject: IDataObject;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IVTDragManager Interface(see page 704)

10.7.1.3.2 IVTDragManager.DragSource Property

Not documented.

Pascal

```
property DragSource: TBaseVirtualTree;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IVTDragManager Interface(see page 704)

10.7.1.3.3 IVTDragManager.DropTargetHelperSupported Property

Not documented.

Pascal

```
property DropTargetHelperSupported: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IVTDragManager Interface(see page 704)

10.7.1.3.4 IVTDragManager.IsDropTarget Property

Not documented.

Pascal

```
property IsDropTarget: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IVTDragManager Interface(see page 704)

10.7.1.3.5 IVTDragManager.ForceDragLeave Method

Not documented.

Pascal

```
procedure ForceDragLeave; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IVTDragManager Interface(see page 704)

10.7.1.3.6 IVTDragManager.GetDataObject Method

Not documented.

Pascal

```
function GetDataObject: IDataObject; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IVTDragManager Interface(see page 704)

10.7.1.3.7 IVTDragManager.GetDragSource Method

Not documented.

Pascal

```
function GetDragSource: TBaseVirtualTree; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IVTDragManager Interface(see page 704)

10.7.1.3.8 IVTDragManager.GetDropTargetHelperSupported Method

Not documented.

Pascal

```
function GetDropTargetHelperSupported: Boolean; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IVTDragManager Interface(see page 704)

10.7.1.3.9 IVTDragManager.GetIsDropTarget Method

Not documented.

Pascal

```
function GetIsDropTarget: Boolean; stdcall;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IVTDragManager Interface(see page 704)

10.7.1.4 IVTEditLink Interface

Interface which is used for communication between the treeview and a node editor.

Pascal

```
IVTEditLink = interface;
```

Description

Due to the virtual nature of the tree it is necessary to supply a kind of plug in interface for application defined node editors. TCustomVirtualStringTree(see TCustomVirtualStringTree Class, page 274) is the first class which implements a node editor. This is just a generic editor to edit a node's caption just like TTreeview does it. Because of the lack of

support under Win9x system this editor only can edit ANSI text. You have to create an own editor to make also Unicode string editing available for node captions.

All node editors must implement this interface to allow the treeview to communicate with the node editor. Node editors are small components or forms. If a node shall be edited (for instance when the user presses F2) the treeview will fire the event OnCreateEditor. The application must determine which node editor must be used for the data in the given node and column. Then it creates and returns an instance of the appropriate node editor.

The life cycle of the node editor object is handled via reference counting. This means that the application must not destroy the node editor explicitly - this will happen automatically when the node editor is not used anymore.

Group

Symbol Reference(see page 701)

Methods

BeginEdit(see IVTEditLink.BeginEdit Method , page 708)
This function will be called by the virtual tree when the editing starts.

CancelEdit(see IVTEditLink.CancelEdit Method , page 709)
This function will be called by the virtual tree when the current editing is about to be cancelled.

EndEdit(see IVTEditLink.EndEdit Method , page 709)
This function will be called by the virtual tree when the current editing is being finished.

GetBounds(see IVTEditLink.GetBounds Method , page 710)
The virtual tree can use this function to get the current bounding rect of the node editor.

PrepareEdit(see IVTEditLink.PrepareEdit Method , page 710)
This function is called by a virtual tree to initialize the node editor.

ProcessMessage(see IVTEditLink.ProcessMessage Method , page 710)
This function is used to forward messages being directed to the virtual tree.

SetBounds(see IVTEditLink.SetBounds Method , page 711)
The virtual tree calls this function to initialize the bounding rectangle of the node editor.

Legend

public

Method

Class Hierarchy

IVTEditLink

File

VirtualTrees

10.7.1.4.1 IVTEditLink.BeginEdit Method

This function will be called by the virtual tree when the editing starts.

Pascal

```
function BeginEdit: Boolean; stdcall;
```

Description

Write code to actually display the node editor here. This might be something like Visible := True or Show. The return value should be true if editing can start or false otherwise. Before this function is called PrepareEdit(see IVTEditLink.PrepareEdit Method , page 710) and SetBounds(see IVTEditLink.SetBounds Method , page 711) are executed.

Interface

IVTEditLink Interface(see page 707)

10.7.1.4.2 IVTEditLink.CancelEdit Method

This function will be called by the virtual tree when the current editing is about to be cancelled.

Pascal

```
function CancelEdit: Boolean; stdcall;
```

Description

Hide the node editor here. This might be something like Visible := False or Hide. The return value should be True if the editing can be cancelled. Return false if the node editor is in an internal state which does not allow to cancel the editing right now.

Do not destroy the node editor instance because this will be done implicitly via reference counting.

Notes

If the edited tree is changed during this function, i.e. focus change, node deletion and so on, CancelEdit might be called again by the tree which can lead to access violations. It is therefore advisable to block reentrancy with a boolean variable. Example:

```
function TStringEditLink.CancelEdit: Boolean;
begin
  Result := not FStopping;
  if Result then
  begin
    FStopping := True;
    FEdit.Hide;
    FTree.CancelEditNode;
  end;
end;
```

Interface

IVTEditLink Interface(see page 707)

10.7.1.4.3 IVTEditLink.EndEdit Method

This function will be called by the virtual tree when the current editing is being finished.

Pascal

```
function EndEdit: Boolean; stdcall;
```

Description

Hide the node editor here. This might be something like Visible := False or Hide. The return value should be true if the editing can be finished. Return false if the node editor is in an internal state which does not allow to finish the editing right now - possibly because there is no valid value available at the moment. If the editing can be finished transmit the edited value to the tree or to the data structure which is displayed in the tree.

Do not destroy the node editor instance because this will be done implicitly via reference counting.

Notes

If the edited tree is changed during this function, i.e. focus change, node deletion and so on, EndEdit might be called

again by the tree which can lead to access violations. It is therefore advisable to block reentrancy with a boolean variable. Example:

```
function TStringEditLink.EndEdit: Boolean;
begin
  Result := not FStopping;
  if Result then
  try
    FStopping := True;
    if FEdit.Modified then
      FTree.DoNewText(FNode, FColumn, FEdit.Caption);
    FEdit.Hide;
  except
    FStopping := False;
    raise;
  end;
end;
```

Interface

IVTEditLink Interface(see page 707)

10.7.1.4.4 IVTEditLink.GetBounds Method

The virtual tree can use this function to get the current bounding rect of the node editor.

Pascal

```
function GetBounds: TRect; stdcall;
```

Description

The bounding rect of the node editor may change during the editing to reflect the changed edit contents. The tree uses this function to query the current bounding rect of the editor. VCL components derived from TControl have a BoundsRect property which can be used as a return value here.

Interface

IVTEditLink Interface(see page 707)

10.7.1.4.5 IVTEditLink.PrepareEdit Method

This function is called by a virtual tree to initialize the node editor.

Pascal

```
function PrepareEdit(Tree: TBaseVirtualTree; Node: PVirtualNode; Column: TColumnIndex): Boolean;
stdcall;
```

Description

Use PrepareEdit to initialize the node editor. This includes getting the node content in the specified column which will be needed later when the editor is shown. BeginEdit(see IVTEditLink.BeginEdit Method , page 708) may be called anytime after this function returns. If the initialization fails simply return false (exceptions should be trapped).

Interface

IVTEditLink Interface(see page 707)

10.7.1.4.6 IVTEditLink.ProcessMessage Method

This function is used to forward messages being directed to the virtual tree.

Pascal

```
procedure ProcessMessage(var Message: TMessage); stdcall;
```

Description

Some node editors might need to trap some messages which are directed to the treeview window. This function remedies the need to subclass the virtual tree via its WindowProc property. If these messages are not needed leave the function body empty.

Interface

IVTEditLink Interface(see page 707)

10.7.1.4.7 IVTEditLink.SetBounds Method

The virtual tree calls this function to initialize the bounding rectangle of the node editor.

Pascal

```
procedure SetBounds(R: TRect); stdcall;
```

Description

This function is usually called after PrepareEdit(see IVTEditLink.PrepareEdit Method , page 710) and before BeginEdit(see IVTEditLink.BeginEdit Method , page 708) in order to place the node editor exactly over the node which is about to be edited. Use the R parameter to set the bounding rect of the editor. If the treeview is in grid mode R will be equal to the cell rectangle of the to be edited cell. Otherwise R is the bounding rectangle of the actual node text.

Notes

SetBounds is also a method of TControl. Hence if your node editor is implemented by a descendant of TControl you must use

a method resolution clause to avoid a name clash. The clause can look similar to this:

```
procedure EditLinkSetBounds(R: TRect); stdcall;
procedure IVTEditLink.SetBounds = EditLinkSetBounds;
```

Interface

IVTEditLink Interface(see page 707)

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X