Introduction

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

What do you think about this topic? Send feedback!
Virtual Treeview is "pure VCL" which means it is not based on any of the system controls but was written from scratch. 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:
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 footprint. 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
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 and IDragSourceHelper 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** 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
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 and CF_VTREFERENCE)*, 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 powerful 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
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**) 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 position 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: DoEventName.

- 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 **powerful 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**: a general purpose procedure to blend a source onto a target bitmap using several different modes.
- **DrawTextW**: a partial implementation of the DrawText API which supports Unicode. This method is not used on Windows NT/2000/XP machines.
- **ShortenString**: 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.

**What do you think about this topic?** Send feedback!
Installation

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

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.

What do you think about this topic? Send feedback!
Version history

Version 4.3.0 - 4.4.2 (December 2004 - November 2005)

- Improvement: fixed column implementation completed (code donation by Igor Savkic)
- 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 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
- 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 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.
- 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 invisible 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 and NeedToUninitialize 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 asynchronously. 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 OnGetCellsIsEmpty
• Improvement: TVTColors.HeaderHotColor introduced, default value is cBtnShadow 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 promoted to public.
• Improvement: ShortenString 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: OnHeaderDbClick 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 alignment was not correctly considered in UpdateEditBounds when grid extensions were enabled.
- Improvement: Check for nil hint data in `TVirtualTreeHintWindow.CalcHintRect` just to be on the safe side.
- Improvement: `TVirtualTreeColumn.ComputeHeaderLayout` 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 `TVTHHeader.AutoFitColumns` 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` 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` 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 public methods are now virtual.
- Change: A couple more methods in the header and columns are virtual now.
- Improvement: Introduction of TVirtualTreeColumnsClass and GetColumnsClass in TVTHeader. This allows for more customization.
- Improvement: DetermineHiddenChildrenFlagAllNodes, tsUpdateHiddenChildrenNeeded, Optimized flag determination to speed up mass changes of the visibility 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` 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 definition 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 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 TApplication.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 introduced to allow descendants to avoid writing columns to the DFM.
- **Renamed Canvas to TargetCanvas in TVTBeforeItemEraseEvent**

(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.**
- **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 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 and TVTNodeCopyingEvent 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.
• 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 erroneously 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 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 explicity 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 (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 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 compatibility
- 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 check images work now also with a themed tree
- OnGetCursor, OnGetHeaderCursor, TVTGetCursorEvent, TVTGetHeaderCursorEvent, 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, TColumnPosition (to utilize better type checking)
- overloaded ColumnFromPosition variant to get a column index from a position index
- TVTHeaderDraggedEvent, 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
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, TVTHeaderStyle, TVTButtonFillMode, XPImages, 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 (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** 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
- 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, ButtonFillMode
• alpha blended selection rectangle
• properties DrawSelectionMode, SelectionRectangleBlendColor and SelectionRectangleBorderColor
• OnHeaderDragged published
• removed TVTEdit.WMKillFocus
• TCustomStringTreeOptions
• 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
• 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 improvements and bug fixes
- changed CF_RTF* to CF_VRTF* 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 comparison
- tree column classes can now be changed by descendants, see
TBaseVirtualTree.GetColumnClass

- **TVTRenderOLEDataEvent**, property `OnRenderOLEData`, `DoRenderOLEData` `TVTGetUserClipboardFormats`, `DoGetUserClipboardFormats`, property `OnGetUserClipboardFormats`
- removed `ScrollIntoView` from `AddChild` and `InsertNode`
- property `OnPaintBackground`, `DoPaintBackground`
- **TVirtualTreeColumn.LoadFromStream** 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` 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
- **TVTDaDataObject**, to have two instances (one for dragging, one for clipboard)
- `IDataObject` is no longer implemented by the drag manager
- renamed `TVTMoveRestriction` to `TVTDragMoveRestriction`
- 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`
  - 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` to have access to the internal drag image bitmap
- `TVTNodeAlignment`, 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, TVTIncrementalSearch, SearchTimer, event OnIncrementalSearch, DoIncrementalSearch, TVTIncrementalSearchEvent, TVTSearchStart, 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, TVTHeader.RestoreColumns
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
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, tsOLEDraggig
- limited auto scroll to draw selection and dragging
- AutoFitColumns
- public property EditLink
- ProcessMessage in IVTEditLink
- improved change handling
- InvalidateColumn
- TVirtualTreeColumns.IsValidColumn
- 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`, `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`
- 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**
- alpha blending
- **Watcher** (critical section) introduced
- MMX **AlphaBlend** implementation
- improved image painting (ghosted, overlay etc.)
- hoDblClickResize
- **TVirtualTreeColumns.AnimatedResize**
- column resize on double click
- GetMaxColumnWidth
- poDrawFocusRect, poDrawSelection in paint options
- ChildNodesOnly in IterateSubtree
- OnColumnClick, OnColumnDbIClick
- HandleMouseDbIClick, WM_RBUTTONDOWNDBLCLK, WM_MBUTTONDOWNDBLCLK
- 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**
- tsPainting state
- simple **DrawTextW** implementation (works also on Win9x)
- improved selection rect painting
- tsValidationNeeded
- check event rework
- PrepareGridExtensions
- CM_ENABLEDCHANGED for design time
- public header click index
- draw selection improvement for all text alignments and bidi modes
- more header mouse events (OnHeaderDbIClick, OnHeaderMouseDown, OnHeaderMouseMove, OnHeaderMouseUp)
- virtual event trigger methods for those mouse events
- multiline hints
- **THitPositions**
- other small changes
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
• 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 declaration for Delphi 4
• TChangeReason 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` 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`
- 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` 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 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 improvements
- 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 columns
• 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, 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/collapses on node focus 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...)

What do you think about this topic? Send feedback!
The virtual paradigm

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). 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) 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 rescanning 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

Links
Inner fundamentals

*What do you think about this topic? Send feedback!*
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.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>This stage is entered only once per paint cycle. After</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>stage</td>
<td>description</td>
<td>details</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>before</td>
<td>painting</td>
<td>setting the vsPainting state it is the very first instruction in a cycle.</td>
</tr>
<tr>
<td>paint</td>
<td>setting the vsPainting state it is the very first instruction in a cycle.</td>
<td>setting another clipping region. Since the passed canvas is not directly used to do 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.</td>
</tr>
<tr>
<td>before</td>
<td>item paint</td>
<td>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. If 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.</td>
</tr>
<tr>
<td>item</td>
<td>paint</td>
<td>This stage 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 in particular the colors are set by the tree according to specific rules (focus, selection etc.).</td>
</tr>
<tr>
<td>item</td>
<td>erase</td>
<td>This stage is also entered only once per node and allows to customize the node's background.</td>
</tr>
<tr>
<td>erase</td>
<td></td>
<td>This stage and its associated event is 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.</td>
</tr>
<tr>
<td>after</td>
<td></td>
<td>This stage and its associated event is used</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
<td>Notes</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>Item erase</td>
<td>only once per node.</td>
<td>to do additional drawings after the background has been erased.</td>
</tr>
<tr>
<td>Before cell paint</td>
<td>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.</td>
<td>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 <em>item erase</em> is that the painting is limited to the current column. There are still no lines or images painted yet.</td>
</tr>
<tr>
<td>On paint text</td>
<td>After default stuff like lines and images has been painted the paint node/paint text stage is entered.</td>
<td>Because Virtual Treeview does not know 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 <em>TVirtualDrawTree</em> simply triggers the OnDrawNode event while the <em>TVirtualStringTree</em> 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. <strong>Note:</strong> The string tree triggers the OnGetText event two times if <em>ToShowStaticText</em> is enabled in the <em>TVirtualStringTree.TreeOptions.StringOptions</em>.</td>
</tr>
</tbody>
</table>
property. Once for the normal text and once for the static text. Use the event's parameter to find out what is required.

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>after cell paint</td>
<td>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.</td>
</tr>
<tr>
<td>after item paint</td>
<td>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.</td>
</tr>
<tr>
<td>after paint</td>
<td>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 (relate the current update area) has been drawn including the selection rectangle.</td>
</tr>
</tbody>
</table>

**Group**

Inner fundamentals

**Links**

Inner fundamentals

*What do you think about this topic? Send feedback!*
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). 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

**Links**

- Inner fundamentals

---

*What do you think about this topic? Send feedback!*
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 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` 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** 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**

**Links**
- **Inner fundamentals**

What do you think about this topic? Send feedback!
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 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
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 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 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

**Links**

Inner fundamentals

*What do you think about this topic? Send feedback!*
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.

<table>
<thead>
<tr>
<th>Key</th>
<th>Modifier</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>none</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>shift</td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>control</td>
<td>Scrolls the tree to the top left corner without change of any selection or focused state.</td>
</tr>
<tr>
<td>End</td>
<td>none</td>
<td>Selects the last visible node (the focused column does not change). This node also receives the input focus. Modifications in grid mode: The focused</td>
</tr>
<tr>
<td>Key Combination</td>
<td>Action Description</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Shift</strong></td>
<td>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. <strong>Modifications in grid mode:</strong> Not the focused node is changed but the last visible column is focused instead. The selection does not change.</td>
<td></td>
</tr>
<tr>
<td><strong>Control</strong></td>
<td>Scrolls the tree to the bottom right corner without change of selection or focused node.</td>
<td></td>
</tr>
<tr>
<td><strong>Prior (page up)</strong></td>
<td>Scrolls the tree window and single selects a node one page up. This node receive also the current focus.</td>
<td></td>
</tr>
<tr>
<td><strong>Shift</strong></td>
<td>Like without modifier key but includes a page of nodes into the current selection.</td>
<td></td>
</tr>
<tr>
<td><strong>Control</strong></td>
<td>Scrolls the tree window one page up without change of selection or focused node.</td>
<td></td>
</tr>
<tr>
<td><strong>Next (page down)</strong></td>
<td>Same as prior but one page down instead.</td>
<td></td>
</tr>
<tr>
<td><strong>Shift</strong></td>
<td>Same as prior but one page down instead.</td>
<td></td>
</tr>
<tr>
<td><strong>Control</strong></td>
<td>Same as prior but one page down instead.</td>
<td></td>
</tr>
<tr>
<td><strong>Up</strong></td>
<td>Advances the focus from the currently focused node to the previous visible node.</td>
<td></td>
</tr>
<tr>
<td><strong>Shift</strong></td>
<td>Advances the focus and adds the newly focused node to the current selection.</td>
<td></td>
</tr>
<tr>
<td><strong>Control</strong></td>
<td>Scrolls the tree window one line up. One line is defined as the DefaultNodeHeight.</td>
<td></td>
</tr>
<tr>
<td><strong>Down</strong></td>
<td>Same as up but one line down instead.</td>
<td></td>
</tr>
<tr>
<td>Modifier</td>
<td>Key</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Left</td>
<td>shift</td>
<td>Same as <strong>up</strong> but one line down instead.</td>
</tr>
<tr>
<td>Left</td>
<td>control</td>
<td>Same as <strong>up</strong> but one line down instead.</td>
</tr>
<tr>
<td>Left</td>
<td>none</td>
<td>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. <strong>Modifications in grid mode:</strong> If extended focus is enabled (see toExtendedFocus in Options) then the behavior changes to a simple navigation to the previous visible column.</td>
</tr>
<tr>
<td>Right</td>
<td>shift</td>
<td>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.</td>
</tr>
<tr>
<td>Right</td>
<td>control</td>
<td>The tree window is scrolled to the left by the amount pixel given in the indent property.</td>
</tr>
<tr>
<td>Right</td>
<td>none</td>
<td>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. <strong>Modifications in grid mode:</strong> If extended focus is enabled (see toExtendedFocus in Options) then the behavior changes to a simple navigation to the next visible column.</td>
</tr>
<tr>
<td>Right</td>
<td></td>
<td>Same as the none-modifier case but the</td>
</tr>
<tr>
<td>Key</td>
<td>Shift</td>
<td>Control</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Description</td>
<td>Selection is extended with the first child node.</td>
<td>Same as <strong>left</strong> but the tree window is scrolled to the right.</td>
</tr>
<tr>
<td>Modifier Keys</td>
<td>Shift</td>
<td>Control</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Meaning</td>
<td>Modifier keys have no meaning for this case.</td>
<td>Modifier keys have no meaning for this case.</td>
</tr>
<tr>
<td>Modifier</td>
<td>Modifier Key</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>+</td>
<td>None</td>
<td>Expands the currently focused node.</td>
</tr>
<tr>
<td></td>
<td>Shift</td>
<td>This modifier alone has no effect, but see the following comment.</td>
</tr>
<tr>
<td></td>
<td>Control</td>
<td>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.</td>
</tr>
<tr>
<td>-</td>
<td>None</td>
<td>Collapses the currently focused node.</td>
</tr>
<tr>
<td></td>
<td>Shift</td>
<td>This modifier alone has no effect, but see the following comment.</td>
</tr>
<tr>
<td></td>
<td>Control</td>
<td>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.</td>
</tr>
<tr>
<td>*</td>
<td>None</td>
<td>Expands the currently focused node and all its children and grand children.</td>
</tr>
<tr>
<td></td>
<td>Shift</td>
<td>This modifier has no effect.</td>
</tr>
<tr>
<td></td>
<td>Control</td>
<td>This modifier has no effect.</td>
</tr>
<tr>
<td>/</td>
<td>None</td>
<td>Collapses the currently focused node and all its children and grand children.</td>
</tr>
<tr>
<td></td>
<td>Shift</td>
<td>This modifier has no effect.</td>
</tr>
<tr>
<td></td>
<td>Control</td>
<td>This modifier has no effect.</td>
</tr>
<tr>
<td>Escape</td>
<td>None</td>
<td>Stops actions which require a specific state in the tree like editing, mouse selecting, drag'n drop etc.</td>
</tr>
<tr>
<td></td>
<td>Shift</td>
<td>This modifier has no effect.</td>
</tr>
<tr>
<td></td>
<td>Control</td>
<td>This modifier has no effect.</td>
</tr>
<tr>
<td>Modifier</td>
<td>Action</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>Space</td>
<td>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. This modifier has no effect.</td>
<td></td>
</tr>
<tr>
<td>Apps (menu key)</td>
<td>Similar to F1 triggers the apps key popup menus on a node by node basis. For more information see also the event OnGetPopupMenu. This modifier has no effect.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>This is the only &quot;normal&quot; character used as hotkey so far. It has only an effect together with the control key. This modifier has no effect.</td>
<td></td>
</tr>
</tbody>
</table>

### 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 view...
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

Links
   Inner fundamentals

What do you think about this topic? Send feedback!
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 preferred 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` 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**

**Links**

**Inner fundamentals**

*What do you think about this topic? Send feedback!*
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

**Links**

Inner fundamentals

*What do you think about this topic? Send feedback!*
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

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.

2001 The parts in this guidance beyond the text from the online help are copyrighted. Every publication requires my admission.

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.

```pascal
  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
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
  // 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.
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

```pascal
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:

```pascal
procedure TMyForm.FormCreate(Sender: TObject);
var
    Node: PVirtualNode;
begin
    ...
    // create tree
    MyTree.NodeDataSize := SizeOf(TTreeData);
    if MyForm.filename = '' then begin // if there is
        // create tree with top level node
        Node := BookmarkForm.BookmarkTree.AddChild(nil)
    end
    else
```
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.

```pascal
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: Node: PVirtualNode; Column: Integer; TextType: TV
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 ikState) is -1.

```pascal
procedure TBookmarkForm.BookmarkTreeGetImageIndex(Sender: PVirtualNode; Kind: TVTImageKind; Column: Integer; var NodeData: ^rTreeData);
begin
  NodeData := Sender.GetNodeData(Node);
  case Kind of
    ikState: // for the case the state icon has been
```
Index := -1;
    ikNormal, ikSelected: // normal or the selected
        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
        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

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

  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;
```
implementation

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

constructor TBasicNodeData.Create(vName: ShortString; vImageIndex: Integer = 0);
begin
  cName := vName;
  cImageIndex := vImageIndex;
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
    // assign data for this node
    NodeD := MyTree.GetNodeData(Node);
    NodeD.BasicND := TBasicNodeData.Create('new pro
end
else
begin
    // load tree
end;

// returns the text (the identification) of the node
procedure TMyForm.MyTreeGetText(Sender: TBaseVirtual
    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: TBase
    Node: PVirtualNode; Kind: TVTImageKind; Column:
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;

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:

```pascal
function TBasicNodeData.GetImageIndex(focus: Boolean): Integer;
begin
  if focus then
    Result := cImageIndexFocus
  else
    Result := cImageIndex;
end;
```

where `cImageIndex` has always the normal index and `cImageIndexFocus` 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:

```pascal
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:

```pascal
procedure TMyForm.addClick (Sender: TObject);
var
  Node: PVirtualNode;
  NodeD: ^rTreeData;
begin
  // Ok, a node must be added.
  Node := MyTree.AddChild(MyTree.FocusedNode); // a
  // 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 the node with the focus must be deleted the following happens:

```pascal
procedure TMyForm.delClick (Sender: TObject);
begin
  // The focused node should be removed. The top level must not be deleted.
  if MyTree.FocusedNode = nil then
    MessageDlg('There was no node selected.', mtInformation, [mbOk], 0)
  else
    // Note: RootNode is the internal (hidden) root node
    // for all top level items.
```
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:

```pascal
procedure TMyForm.MyTreeFreeNode(Sender: TBaseVirtualTree); begin
    // Free here the node data (Note: type PtreeData
```
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

```pascal
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.

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

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

```pascal
procedure TMyForm.MyTreeSaveNode(Sender: TBaseVirtualTree);
begin
```

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

```pascal
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
  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:

```pascal
try
  PTreeData(Sender.GetNodeData(Node)).BasicND.SaveToFile(Stream);
end;
```
By the call of LoadFromFile the event OnLoadNode will be triggered and consequently the method LoadNode:

```pascal
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; 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
    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; Kind: TVTImageKind; Column: Integer; var Index: Integer);

var
  NodeD: ^rTreeData;

begin
  NodeD := Sender.GetNodeData(Node);

  if Column = 0 then // icons only in the first col
    begin
      case Kind of
        ikState:
        Index := -1;
      end;
    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:

```pascal
procedure TBookmarkForm.BookmarkTreeColumnResize(Sender: TObject);
var
  NodeD: PTreeData;
begin
  NodeD := Sender.GetNodeData(Sender.RootNode.FirstChild);
  // Keep the new size of the column in the project
  TVirtualStringTree(Sender).Header.Columns.Items[Column].Width,
  TProjectNodeData(NodeD.BasicND).SetHColumnsWidth
end;
```

The exciting part is the type casting of the sender object. In
**TBaseVirtualTree** 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...

What do you think about this topic? Send feedback!
A little code repository by John Knipper

This is just written by me John Knipper. 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:

```cpp
VirtualStringTree1.RootNodeCount := 5; // is adding
```

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:

```cpp
var
  Node: PVirtualNode;
Node := VirtualStringTree1.AddChild(nil); // Adds
Node := VirtualStringTree1.AddChild(ParentNode);
Node := VirtualStringTree1.InsertNode(Node, amIns
```

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:

```pascal
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);
```

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; 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
    else Level := 0;
  end;
end;
```
case FFillMode of
  0: // fill tree with a specific amount of nodes
    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 there);
  2: // fill tree with a certain amount of root
    begin
      Data.FixedText := True;
    end;
  3: // fill tree with a certain amount of root
    // up to an absolute amount of ~1 million
    begin
      if Assigned(ParentNode) then Count := ParentNode.ChildCount
      else Count := TVirtualStringTree
        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, ivsHasChildren);
  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

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

Or in that case you can use

```pascal
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:

```pascal
procedure TFRM_WWW_main.vFavTreeGetText(Sender: TBaseVirtualTree)
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 =
  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:**

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

**Q:** How can I display text for other columns?

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

**E:**

```pascal
procedure TFRM_WWW_main.vFavTreeGetText(Sender: TBaseVirtualTree)
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 =
  begin
    Table.GotoBookmark(TBookmark(Sender.GetNodeData(Node)));
    Text := Table.FieldByName('Name').asString;
  end;
end;
```
begin
  case Column of
    -1, // main column, -1 if columns are hidden, 0:
      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:

begin
  if Kind = ikState then
  begin
    Index := 2;
  end
  else
    if (Kind = ikNormal) or (Kind = ikSelected) the
    begin
      Index := 1;
    end;
end;

or just use
procedure TFRM_WWW_main.vFavTreeGetImageIndex(Sender:
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.

What do you think about this topic? Send feedback!
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. 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 InvalidateNode 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) 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 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 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.

What do you think about this topic? Send feedback!
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. 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 maintained by

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

What do you think about this topic? Send feedback!
These are all classes that are contained in this documentation.

Classes

- EVirtualTreeError
- TBaseVirtualTree
  - TBaseVirtualTree is the main and base class for all other Virtual Treeview descendants.
- TBufferedString
- TClipboardFormatList
  - Not documented.
- TClipboardFormats
  - List of strings describing clipboard formats.
- TCriticalSection
  - Not documented.
- TCustomStringTreeOptions
  - Enhanced options class for string trees.
- TCustomVirtualDrawTree
  - Simple owner draw descendant of the base tree.
- TCustomVirtualStringTree
  - Descendant of TBaseVirtualTree, which is able to manage node captions on its own
- TCustomVirtualTreeOptions
  - Organizes all tree options into subproperties for easier management.
- TEnumFormatEtc
- TScrollBarOptions
- **TStringEditLink**
  TStringEditLink is the standard node editor of a *TVirtualStringTree*.

- **TStringTreeOptions**
  Options class used in the string tree and its descendants.

- **TVirtualDrawTree**
  Descendant of *TBaseVirtualTree*, which passes node paint events through to the application (similar to a draw grid).

- **TVirtualStringTree**
  Descendant of *TBaseVirtualTree* which is able to manage node captions on its own.

- **TVirtualTreeColumn**
  Represents a column in a Virtual Treeview.

- **TVirtualTreeColumns**
  Collection class, which holds the columns for the tree.

- **TVirtualTreeHintWindow**
  Internally used hint window class to support Unicode hints.

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

- **TVTColors**
  Collects all color related options for the tree control.

- **TVTDataObject**
  Implementation of an IDataObject interface.

- **TVTDragImage**
  Not documented.

- **TVTDragManager**
  Not documented.

- **TVTEdit**
  Not documented.

- **TVTHeader**
  Not documented.

- **TVTHeaderPopupMenu**
  Not documented.

- **TWideBufferedString**
  Not documented.
TWorkerThread
Not documented.

TWriterHack
Not documented.

Group
Virtual Treeview

Legend

Class

Links
Classes, Virtual Treeview, Legend

What do you think about this topic? Send feedback!
**EVirtualTreeError Class**

**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.
TBaseVirtualTree Class

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

Pascal

```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 publish. Do not use the base treeview as object. It is not meant to be instantiated directly, instead via an descendant.

Group

Classes

Members

Properties

- **Alignment**
  Determines the horizontal alignment of text if no columns are defined.
- **AnimationDuration**
  Determines the maximum duration the tree can use to play an animation.
- **AutoExpandDelay**
  Time delay after which a node gets expanded if it is the current drop
AutoScrollDelay
Time which determines when auto scrolling should start.

AutoScrollInterval
Time interval between scroll events when doing auto scroll.

Background
Holds a background image for the tree.

BackgroundOffsetX
Horizontal offset of the background image.

BackgroundOffsetY
Vertical offset of the background image.

BorderStyle
Same as TForm.BorderStyle.

ButtonFillMode
Determines how to fill the background of the node buttons.

ButtonStyle
Determines the look of node buttons.

ChangeDelay
Time which determines when the OnChange event should be triggered after the actual change event.

CheckImageKind
Determines which images should be used for checkboxes and radio buttons.

CheckImages
Not documented.

CheckState
Read or set the check state of a node.

CheckType
Read or set the check type of a node.

ChildCount
Read or set the number of child nodes of a node.

ChildrenInitialized
Read whether a node's child count has been initialized already.

ClipboardFormats
Special class to keep a list of clipboard format descriptions.
- **Colors**
  A collection of colors used in the tree.

- **CustomCheckImages**
  Assign your own image list to get the check images you like most.

- **DefaultNodeHeight**
  Read or set the height new nodes get as initial value.

- **DefaultPasteMode**
  Read or set the value, which determines where to add pasted nodes to.

- **DragHeight**
  Read or set the vertical limit of the internal drag image.

- **DragImage**
  Holds the instance of the internal drag image.

- **DragImageKind**
  Read or set what should be shown in the drag image.

- **DragManager**
  Holds the reference to the internal drag manager.

- **DragOperations**
  Read or set which drag operations may be allowed in the tree.

- **DragSelection**
  Keeps a temporary list of nodes during drag'n drop.

- **DragType**
  Read or set which subsystem should be used for dragging.

- **DragWidth**
  Read or set the horizontal limit of the internal drag image.

- **DrawSelectionMode**
  Read or set how multiselection with the mouse is to be visualized.

- **DropTargetNode**
  Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

- **EditColumn**
  Not documented.

- **EditDelay**
  Read or set the maximum time between two single clicks on the same node, which should start node editing.
**EditLink**
Keeps a reference to the internal edit link during a node edit operation.

**Expanded**
Read or set the expanded state of a particular node.

**FocusedColumn**
Read or set the currently focused column.

**FocusedNode**
Read or set the currently focused node.

**Font**
Same as TWinControl.Font.

**FullyVisible**
Read or set whether a node is fully visible or not.

**HasChildren**
Read or set whether a node has got children.

**Header**
Provides access to the header instance.

**HeaderRect**
Returns the non-client-area rectangle used for the header.

**HintAnimation**
Read or set the current hint animation type.

**HintMode**
Read or set what type of hint you want for the tree view.

**HotCursor**
Read or set which cursor should be used for hot nodes.

**HotNode**
Read, which node is currently the hot node.

**Images**
Read or set the tree's normal image list.

**IncrementalSearch**
Read or set the current incremental search mode.

**IncrementalSearchDirection**
Read or set the direction to be used for incremental search.

**IncrementalSearchStart**
Read or set where to start incremental search.
**IncrementalSearchTimeout**
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

**Indent**
Read or set the indentation amount for node levels.

**IsDisabled**
Read or set the enabled state of the given node.

**IsVisible**
Read or set the visibility state of the given node.

**LastClickPos**
Used for retained drag start and wheel mouse scrolling.

**LastDropMode**
Read how the last drop operation finished.

**LineMode**
Read or set the mode of the tree lines.

**LineStyle**
Read or set the mode of the tree lines.

**Margin**
Read or set the tree's node margin.

**MultiLine**
Read or toggle the multiline feature for a given node.

**NodeAlignment**
Read or set the node alignment value.

**NodeDataSize**
Read or set the extra data size for each node.

**NodeHeight**
Read or set a node's height.

**NodeParent**
Read or set a node's parent node.

**OffsetX**
Read or set the tree's current horizontal and vertical scroll offsets.

**OffsetXY**
Read or set the tree's current horizontal and vertical scroll offsets.

**OffsetY**
Read or set the tree's current horizontal and vertical scroll offsets.
**RootNode**
Reference to the internal root node which is the anchor of the entire tree node hierarchy.

**RootNodeCount**
Read or set the number of nodes on the top level.

**ScrollBarOptions**
Reference to the scroll bar options class.

**SearchBuffer**
Current input string for incremental search.

**Selected**
Property to modify or determine the selection state of a node.

**SelectedCount**
Contains the number of selected nodes.

**SelectionBlendFactor**
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

**SelectionCurveRadius**
Read or set the current corner radius for node selection rectangles.

**StateImages**
Reference to the images list which is used for the state images.

**TextMargin**
Read or set the distance of the node caption to its borders.

**TopNode**
The top node is the node which is currently at the top border of the client area.

**TotalCount**
Returns the number of nodes in the tree.

**TotalInternalDataSize**
Keeps the currently accumulated data size for one node.

**TreeOptions**
Reference to the tree's options.

**TreeStates**
Property which keeps a set of flags which indicate current operation and states of the tree.

**UpdateCount**
Not documented.

**VerticalAlignment**
Used to set a node's vertical button alignment with regard to the entire node rectangle.

**VisibleCount**
Number of currently visible nodes.

**VisiblePath**
Property to set or determine a node parent's expand states.

**WantTabs**
Read or set whether the tree wants to process tabs on its own.

**Events**

- **OnAdvancedHeaderDraw**
  Header paint support event.

- **OnAfterCellPaint**
  Paint support event.

- **OnAfterItemErase**
  Paint support event.

- **OnAfterItemPaint**
  Paint support event.

- **OnAfterPaint**
  Paint support event.

- **OnBeforeCellPaint**
  Paint support event.

- **OnBeforeItemErase**
  Paint support event.

- **OnBeforeItemPaint**
  Paint support event.

- **OnBeforePaint**
  Paint support event.

- **OnChange**
  Navigation support event.

- **OnChecked**
  Check support event.
OnChecking
Check support event.

OnCollapsed
Miscellaneous event.

OnCollapsing
Miscellaneous event.

OnColumnClick
Header and column support event.

OnColumnDbClick
Header and column support event.

OnColumnResize
Header and column support routine.

OnCompareNodes
Sort and search support event.

OnCreateDataObject
Drag'n drop support event.

OnCreateDragManager
Drag'n drop support event.

OnCreateEditor
Editing support event.

OnDragAllowed
Drag'n drop support event.

OnDragDrop
Drag'n drop support event.

OnDragOver
Drag'n drop support event.

OnEditCancelled
Editing support event.

OnEdited
Editing support event.

OnEditing
Editing support event.

OnExpanded
Miscellaneous event.

OnExpanding
Miscellaneous event.
- **OnFocusChanged**
  Navigation support event.
- **OnFocusChanging**
  Navigation support event.
- **OnFreeNode**
  Data management node.
- **OnGetCellIsEmpty**
  Triggered when the tree control needs to know whether a given column is empty.
- **OnGetCursor**
  Miscellaneous event.
- **OnGetHeaderCursor**
  Header and column support event.
- **OnGetHelpContext**
  Miscellaneous event.
- **OnGetImageIndex**
  Display management event.
- **OnGetImageIndexEx**
  Not documented.
- **OnGetLineStyle**
  Display management event.
- **OnGetNodeDataSize**
  Data management event.
- **OnGetPopupMenu**
  Miscellaneous event.
- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.
- **OnHeaderClick**
  Header & column support event.
- **OnHeaderDblClick**
  Header & column support event.
- **OnHeaderDragged**
  Header & column support event.
- **OnHeaderDraggedOut**
Header & column support event.
- **OnHeaderDragging**
  Header & column support event.
- **OnHeaderDraw**
  Header & column support event.
- **OnHeaderDrawQueryElements**
  Header & column support event.
- **OnHeaderMouseDown**
  Header & column support event.
- **OnHeaderMouseMove**
  Header & column support event.
- **OnHeaderMouseUp**
  Header & column support event.
- **OnHotChange**
  Navigation support event.
- **OnIncrementalSearch**
  Miscellaneous event.
- **OnInitChildren**
  Node management event.
- **OnInitNode**
  Node management event.
- **OnKeyAction**
  Miscellaneous event.
- **OnLoadNode**
  Streaming support event.
- **OnMeasureItem**
  Miscellaneous event.
- **OnNodeCopied**
  Miscellaneous event.
- **OnNodeCopying**
  Miscellaneous event.
- **OnNodeMoved**
  Miscellaneous event.
- **OnNodeMoving**
  Miscellaneous event.
OnPaintBackground
Paint support event.

OnRenderOLEData
Drag'n drop and clipboard support event.

OnResetNode
Node management event.

OnSaveNode
Streaming support event.

OnScroll
Miscellaneous event.

OnShowScrollbar
Not documented.

OnStateChange
Miscellaneous event.

OnStructureChange
Miscellaneous event.

OnUpdating
Miscellaneous event.

Methods

AbsolutePath
Reads the overall index of a node.

AddChild
Creates and adds a new child node to given node.

AddFromStream
Adds the content from the given stream to the given node.

AddToSelection
Adds one or more nodes to the current selection.

AdjustPaintCellRect
Used in descendent to modify the clip rectangle of the current column while painting a certain node.

AdjustPanningCursor
Loads the proper cursor which indicates into which direction scrolling is done.
AdviseChangeEvent
Used to register a delayed change event.

AllocateInternalDataArea
Registration method to allocate tree internal data per node.

Animate
Support method for animated actions in the tree view.

Assign
Used to copy properties from another Virtual Treeview.

BeginDrag
Starts an OLE drag'n drop operation.

BeginSynch
Enters the tree into a special synchronized mode.

BeginUpdate
Locks the tree view to perform several update operations.

CalculateSelectionRect
Support method for draw selection.

CanAutoScroll
Determines whether the tree can currently auto scroll its window.

CancelCutOrCopy
Cancels any pending cut or copy clipboard operation.

CancelEditNode
Cancel the current edit operation, if there is any.

CanEdit
Determines whether a node can be edited or not.

CanFocus
Support method to determine whether the tree window can receive the input focus.

CanShowDragImage
Determines whether a drag image should be shown.

Change
Central method called when a node's selection state changes.

ChangeScale
Helper method called by the VCL when control resizing is due.

CheckParentCheckState
Helper method for recursive check state changes.
Clear
Clears the tree and removes all nodes.

ClearChecked
Not documented.

ClearSelection
Removes all nodes from the current selection.

ClearTempCache
Helper method to clear the internal temporary node cache.

ColumnIsEmpty
Used to determine if a cell is considered as being empty.

CopyTo
Copies Source and all its child nodes to Target.

CopyToClipboard
Copies all currently selected nodes to the clipboard.

CountLevelDifference
Determines the level difference of two nodes.

CountVisibleChildren
Determines the number of visible child nodes of the given node.

Create
Constructor of the control

CreateParams
Prepares the creation of the controls window handle.

CreateWnd
Initializes data, which depends on the window handle.

CutToClipboard
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties
Helper method to customize loading and saving persistent tree data.

DeleteChildren
Removes all child nodes from the given node.

DeleteNode
Removes the given node from the tree.

DeleteSelectedNodes
Removes all currently selected nodes from the tree.
Destroy
Destructor of the control.

DetermineHiddenChildrenFlag
Determines whether all children of a given node are hidden.

DetermineHiddenChildrenFlagAllNodes
Determines whether all children of all nodes are hidden.

DetermineHitPositionLTR
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineHitPositionRTL
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineNextCheckState
Not documented.

DetermineScrollDirections
Not documented.

DoAdvancedHeaderDraw
Not documented.

DoAfterCellPaint
Not documented.

DoAfterItemErase
Not documented.

DoAfterItemPaint
Not documented.

DoAfterPaint
Not documented.

DoAutoScroll
Enables or disables the auto scroll timer.

DoBeforeCellPaint
Not documented.

DoBeforeDrag
Not documented.

DoBeforeItemErase
Not documented.

DoBeforeItemPaint
Not documented.

DoBeforePaint
Not documented.

DoCancelEdit
Called when the tree should stop editing without accepting changed values.

DoCanEdit
Not documented.

DoChange
Not documented.

DoCheckClick
Not documented.

DoChecked
Not documented.

DoChecking
Not documented.

DoCanEdit
Not documented.

DoCollapsing
Not documented.

DoColumnClick
Not documented.

DoColumnDblClick
Not documented.

DoColumnResize
Not documented.

DoCompare
Not documented.

DoCreateDataObject
Not documented.

DoCreateDragManager
Not documented.

DoCreateEditor
Not documented.

DoDragDrop
Not documented.

- DoDragExpand
  Not documented.
- DoDragging
  Internal method which handles drag' drop.
- DoDragOver
  Not documented.
- DoEdit
  Initiates editing of the currently set focused column and edit node.
- DoEndDrag
  Not documented.
- DoEndEdit
  Stops the current edit operation and takes over the new content.
- DoExpanded
  Not documented.
- DoExpanding
  Not documented.
- DoFocusChange
  Not documented.
- DoFocusChanging
  Not documented.
- DoFocusNode
  Internal method to set the focused node.
- DoFreeNode
  Not documented.
- DoGetAnimationType
  Determines the type of animation to be used.
- DoGetCursor
  Not documented.
- DoGetHeaderCursor
  Not documented.
- DoGetImageIndex
  Not documented.
- DoGetLineStyle
  Not documented.
DoGetNodeHint
  Not documented.
DoGetNodeTooltip
  Not documented.
DoGetNodeWidth
  Overridable method which always returns 0.
DoGetPopupMenu
  Overridable method which triggers the OnGetPopup event.
DoGetUserClipboardFormats
  Not documented.
DoHeaderClick
  Not documented.
DoHeaderDblClick
  Not documented.
DoHeaderDragged
  Not documented.
DoHeaderDraggedOut
  Not documented.
DoHeaderDragging
  Not documented.
DoHeaderDraw
  Not documented.
DoHeaderDrawQueryElements
  Not documented.
DoHeaderMouseDown
  Not documented.
DoHeaderMouseMove
  Not documented.
DoHeaderMouseUp
  Not documented.
DoHotChange
  Not documented.
DoIncrementalSearch
  Not documented.
DoInitChildren
Not documented.

DoInitNode
Not documented.

DoKeyAction
Not documented.

DoLoadUserData
Not documented.

DoMeasureItem
Not documented.

DoNodeCopied
Not documented.

DoNodeCopying
Not documented.

DoNodeMoved
Not documented.

DoNodeMoving
Not documented.

DoPaintBackground
Not documented.

DoPaintDropMark
Overridable method which draws the small line on top of a nodes image depending on the current drop state.

DoPaintNode
Overridable method which does nothing.

DoPopupMenu
Overridable method which shows the popup menu for the given node.

DoRenderOLEData
Not documented.

DoReset
Not documented.

DoSaveUserData
Not documented.

DoScroll
Overridable method which triggers the OnScroll event.

DoSetOffsetXY
Internal core routine to set the tree's scroll position.

- **DoShowScrollbar**
  Not documented.

- **DoStartDrag**
  Not documented.

- **DoStateChange**
  Not documented.

- **DoStructureChange**
  Not documented.

- **DoTimerScroll**
  Callback method which is triggered whenever the scroll timer fires.

- **DoUpdating**
  Not documented.

- **DoValidateCache**
  Not documented.

- **DragCanceled**
  Called by the VCL when a drag'n drop operation was canceled by the user.

- **DragDrop**
  Helper method, which is used when a drag operation is finished.

- **DragEnter**
  Not documented.

- **DragFinished**
  Called when a drag operation is finished (accepted or cancelled).

- **Dragging**
  Returns true if a drag'n drop operation is in progress.

- **DragLeave**
  Not documented.

- **DragOver**
  Not documented.

- **DrawDottedHLine**
  Not documented.

- **DrawDottedVLine**
  Not documented.

- **EditNode**
Starts editing the given node if allowed to.

- **EndEditNode**
  Stops node editing if it was started before.

- **EndSynch**
  Counterpart to **BeginSynch**.

- **EndUpdate**
  Resets the update lock set by **BeginUpdate**.

- **ExecuteAction**
  Not documented.

- **FindNodeInSelection**
  Helper method to find the given node in the current selection.

- **FinishChunkHeader**
  Not documented.

- **FinishCutOrCopy**
  Stops any pending cut or copy clipboard operation.

- **FlushClipboard**
  Renders all pending clipboard data.

- **FontChanged**
  Not documented.

- **FullCollapse**
  Collapses all nodes in the tree.

- **FullExpand**
  Expands all nodes in the tree.

- **GetBorderDimensions**
  Not documented.

- **GetCheckImage**
  Not documented.

- **GetCheckImageListFor**
  Not documented.

- **GetColumnClass**
  Returns the class to be used to manage columns in the tree.

- **GetControlsAlignment**
  Not documented.

- **GetDisplayRect**
  Returns the visible region used by the given node in client
coordinates.

- **GetFirst**
  Group of node navigation functions.

- **GetFirstChecked**
  Not documented.

- **GetFirstChild**
  Group of node navigation functions.

- **GetFirstCutCopy**
  Group of node navigation functions.

- **GetFirstInitialized**
  Group of node navigation functions.

- **GetFirstNoInit**
  Group of node navigation functions.

- **GetFirstSelected**
  Group of node navigation functions.

- **GetFirstVisible**
  Group of node navigation functions.

- **GetFirstVisibleChild**
  Group of node navigation functions.

- **GetFirstVisibleChildNoInit**
  Group of node navigation functions.

- **GetFirstVisibleNoInit**
  Group of node navigation functions.

- **GetHeaderClass**
  Returns the header class to be used by the tree.

- **GetHintWindowClass**
  Not documented.

- **GetHitTestInfoAt**
  Returns information about the node at the given position.

- **GetImageIndex**
  Not documented.

- **GetLast**
  Group of node navigation functions.

- **GetLastChild**
  Group of node navigation functions.
GetFirstChildNoInit
Group of node navigation functions.

GetFirstChildInitialized
Group of node navigation functions.

GetFirstChildNoInit
Group of node navigation functions.

GetFirstChild
Group of node navigation functions.

GetFirstChildVisible
Group of node navigation functions.

GetFirstChildVisibleChild
Group of node navigation functions.

GetFirstChildVisibleChildNoInit
Group of node navigation functions.

GetFirstChildVisibleNoInit
Group of node navigation functions.

GetMaxColumnWidth
Returns the width of the largest node in the given column.

GetMaxRightExtend
Determines the maximum width of the currently visible part of the tree.

GetNativeClipboardFormats
Used to let descendants and the application add their own supported clipboard formats.

GetNext
Group of node navigation functions.

GetNextChecked
Not documented.

GetNextCutCopy
Group of node navigation functions.

GetNextInitialized
Group of node navigation functions.

GetNextNoInit
Group of node navigation functions.

GetNextSelected
Group of node navigation functions.

GetNextSibling
Group of node navigation functions.
GetNextVisible
Group of node navigation functions.

GetNextVisibleNoInit
Group of node navigation functions.

GetNextVisibleSibling
Group of node navigation functions.

GetNextVisibleSiblingNoInit
Group of node navigation functions.

GetNodeAt
Not documented.

GetNodeData
Returns the address of the user data area of the given node.

GetNodeLevel
Returns the indentation level of the given node.

GetOptionsClass
Customization helper to determine which options class the tree should use.

GetPrevious
Group of node navigation functions.

GetPreviousInitialized
Group of node navigation functions.

GetPreviousNoInit
Group of node navigation functions.

GetPreviousSibling
Group of node navigation functions.

GetPreviousVisible
Group of node navigation functions.

GetPreviousVisibleNoInit
Group of node navigation functions.

GetPreviousVisibleSibling
Group of node navigation functions.

GetPreviousVisibleSiblingNoInit
Group of node navigation functions.

GetSortedCutCopySet
Returns a sorted list of nodes, which are marked for cut or copy
clipboard operation.

- **GetSortedSelection**
  Returns a sorted list of all currently selected nodes.

- **GetTextInfo**
  Helper method for node editors, hints etc.

- **GetTreeFromDataObject**
  OLE drag'n drop and clipboard support method.

- **GetTreeRect**
  Returns the size of the virtual tree image.

- **GetVisibleParent**
  Returns the first (nearest) parent node, which is visible.

- **HandleHotTrack**
  Not documented.

- **HandleIncrementalSearch**
  Not documented.

- **HandleMouseDblClick**
  Not documented.

- **HandleMouseDown**
  Not documented.

- **HandleMouseUp**
  Not documented.

- **HasAsParent**
  Determines if the given node has got another node as one of its parents.

- **HasImage**
  Not documented.

- **HasPopupMenu**
  Determines whether there is a pop up menu assigned to the tree.

- **InitChildren**
  Not documented.

- **InitNode**
  Not documented.

- **InsertNode**
  Inserts a new node and returns it to the caller.

- **InternalAddFromStream**
Not documented.

- **InternalAddToSelection**
  Not documented.
- **InternalCacheNode**
  Not documented.
- **InternalClearSelection**
  Not documented.
- **InternalConnectNode**
  Not documented.
- **InternalData**
  Returns the address of the internal data for a tree class.
- **InternalDisconnectNode**
  Not documented.
- **InternalRemoveFromSelection**
  Not documented.
- **InvalidateCache**
  Empties the internal node cache and marks it as invalid.
- **InvalidateChildren**
  Invalidates all children of the given node.
- **InvalidateColumn**
  Invalidates the client area part of a column.
- **InvalidateNode**
  Invalidates the given node.
- **InvalidateToBottom**
  Invalidates the client area starting with the top position of the given node.
- **InvertSelection**
  Inverts the current selection.
- **IsEditing**
  Tells the caller whether the tree is currently in edit mode.
- **IsMouseSelecting**
  Tell the caller whether the tree is currently in draw selection mode.
- **IterateSubtree**
  Iterator method to go through all nodes of a given sub tree.
- **Loaded**
Not documented.

LoadFromFile
Loads previously streamed out tree data back in again.

LoadFromStream
Loads previously streamed out tree data back in again.

MainColumnChanged
Not documented.

MarkCutCopyNodes
Not documented.

MeasureItemHeight
Not documented.

MouseMove
Not documented.

MoveTo
Moves Source and all its child nodes to Target.

Notification
Not documented.

OriginalWMNCPaint
Not documented.

Paint
TControl's Paint method used here to display the tree.

PaintCheckImage
Not documented.

PaintImage
Not documented.

PaintNodeButton
Not documented.

PaintSelectionRectangle
Not documented.

PaintTree
Main paint routine for the tree image.

PaintTreeLines
Not documented.

PanningWindowProc
Not documented.
PasteFromClipboard
Inserts the content of the clipboard into the tree.

PrepareDragImage
Not documented.

Print
Not documented.

ProcessDrop
Helper method to ease OLE drag'n drop operations.

ProcessOLEData
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk
Not documented.

ReadNode
Not documented.

RedirectFontChangeEvent
Not documented.

ReinitChildren
Forces all child nodes of Node to be reinitialized.

ReinitNode
Forces a reinitialization of the given node.

RemoveFromSelection
Removes the given node from the current selection.

RenderOLEData
Renders pending OLE data.

RepaintNode
Causes the treeview to repaint the given node.

ResetNode
Resets the given node to uninitialized.

ResetRangeAnchor
Not documented.

RestoreFontChangeEvent
Not documented.

SaveToFile
Saves the entire content of the tree into a file or stream.

SaveToStream
Saves the entire content of the tree into a file or stream.

ScrollIntoView
Scrolls the tree so that the given node comes in the client area.

SelectAll
Selects all nodes in the tree.

SelectNodes
Selects a range of nodes.

SetBiDiMode
Not documented.

SetFocusedNodeAndColumn
Not documented.

SkipNode
Not documented.

Sort
Sorts the given node.

SortTree
Sorts the entire tree view.

StartWheelPanning
Not documented.

StopWheelPanning
Not documented.

StructureChange
Not documented.

SuggestDropEffect
Not documented.

ToggleNode
Changes a node's expand state to the opposite state.

ToggleSelection
Toggles the selection state of a range of nodes.

UnselectNodes
Deselects a range of nodes.

UpdateAction
Not documented.

UpdateDesigner
Not documented.
UpdateEditBounds
Not documented.
UpdateHeaderRect
Not documented.
UpdateHorizontalScrollBar
Applies changes to the horizontal and vertical scrollbars.
UpdateScrollBars
Applies changes to the horizontal and vertical scrollbars.
UpdateVerticalScrollBar
Applies changes to the horizontal and vertical scrollbars.
UpdateWindowAndDragImage
Not documented.
UseRightToLeftReading
Helper method for right-to-left layout.
ValidateCache
Initiates the validation of the internal node cache.
ValidateChildren
Validates all children of a given node.
ValidateNode
Validates a given node.
ValidateNodeDataSize
Helper method for node data size initialization.
WndProc
Redirected window procedure to do some special processing.
WriteChunks
Writes the core chunks for the given node to the given stream.
WriteNode
 Writes the cover (envelop) chunk for the given node to the given stream.

Legend

protected

Property
public
read only
Event
Method
virtual

Class Hierarchy

File
VirtualTrees

Links
Events, Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TBaseVirtualTree.Alignment Property
TBaseVirtualTree Class

Determines the horizontal alignment of text if no columns are defined.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.AnimationDuration Property

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class
TBaseVirtualTree.AutoExpandDelay Property

TBaseVirtualTree Class

Time delay after which a node gets expanded if it is the current drop target.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.AutoScrollDelay Property

Time which determines when auto scrolling should start.

Pascal

```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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.AutoScrollInterval Property**

*TBaseVirtualTree Class*

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*

**Links**

*TBaseVirtualTree Class*

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.Background Property**

**TBaseVirtualTree Class**

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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.BackgroundOffsetX Property
TBaseVirtualTree Class

Horizontal offset of the background image.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.BackgroundOffsetY Property**

**TBaseVirtualTree Class**

Vertical offset of the background image.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.BorderStyle Property

TBaseVirtualTree Class

Same as TForm.BorderStyle.

**Pascal**

```pascal
property BorderStyle: TBorderStyle;
```

**Description**

See TForm.BorderStyle.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.ButtonFillMode Property

TBaseVirtualTree Class

Determines how to fill the background of the node buttons.

Pascal

```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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.ButtonStyle Property
TBaseVirtualTree Class

Determines the look of node buttons.

Pascal

```pascal
property ButtonStyle: TVTButtonStyle;
```

Description

Determines the look of node buttons.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.ChangeDelay Property**

**TBaseVirtualTree Class**

Time which determines when the **OnChange** event should be triggered after the actual change event.

**Pascal**

```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) happen. 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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class
What do you think about this topic? Send feedback!
TBaseVirtualTree.CheckImageKind Property

TBaseVirtualTree Class

Determines which images should be used for checkboxes and radio buttons.

Pascal

```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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
TBaseVirtualTree.CheckImages Property
TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.CheckState Property

TBaseVirtualTree Class

Read or set the check state of a node.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.CheckType Property

Read or set the check type of a node.

Pascal

```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 the node's check state to csUncheckedNormal.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
TBaseVirtualTree.ChildCount Property

Read or set the number of child nodes of a node.

Pascal

```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, InsertNode and DeleteNode.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
TBaseVirtualTree.ChildrenInitialized Property

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.ClipboardFormats Property

TBaseVirtualTree Class

Special class to keep a list of clipboard format descriptions.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.Colors Property

TBaseVirtualTree Class

A collection of colors used in the tree.

Pascal

```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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.CustomCheckImages Property

Assign your own image list to get the check images you like most.

Pascal

```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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
**TBaseVirtualTree.DefaultNodeHeight Property**

**TBaseVirtualTree Class**

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

**Links**

TBaseVirtualTree Class

---

What do you think about this topic? Send feedback!
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

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
What do you think about this topic? Send feedback!


**TBaseVirtualTree.DragHeight Property**

*TBaseVirtualTree Class*

Read or set the vertical limit of the internal drag image.

**Pascal**

```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**

*TBaseVirtualTree Class*

**Links**

*TBaseVirtualTree Class*

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DragImage Property**

*TBaseVirtualTree Class*

Holds the instance of the internal drag image.

**Pascal**

```
property DragImage: TVTDragImage;
```

**Description**

For older systems where the `IDropTargetHelper` 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` interface is supported.

**Class**

*TBaseVirtualTree Class*

**Links**

*TBaseVirtualTree Class*

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DragImageKind Property**

* TBaseVirtualTree Class

Read or set what should be shown in the drag image.

**Pascal**

```pascal
property DragImageKind: TVTDragImageKind;
```

**Description**

DragImageKind allows to switch parts of the drag image off and on.

**Class**

* TBaseVirtualTree Class

**Links**

* TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DragManager Property

Holds the reference to the internal drag manager.

Pascal

```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

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
**TBaseVirtualTree.DragOperations Property**

**TBaseVirtualTree Class**

Read or set which drag operations may be allowed in the tree.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DragSelection Property

TBaseVirtualTree Class

Keeps a temporary list of nodes during drag'n drop.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DragType Property**

TBaseVirtualTree Class

Read or set which subsystem should be used for dragging.

**Pascal**

```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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DragWidth Property**

**TBaseVirtualTree Class**

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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DrawSelectionMode Property

TBaseVirtualTree Class

Read or set how multiselection with the mouse is to be visualized.

Pascal

```pascal
property DrawSelectionMode: TVTDrawSelectionMode;
```

Description

Virtuall Treeview allows to display two different selection rectangles when doing multiselection with the mouse. One is the traditional 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DropTargetNode

Property

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree>EditColumn** Property

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```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**

**Links**

**TBaseVirtualTree Class**

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.EditDelay Property**

**TBaseVirtualTree Class | See Also**

Read or set the maximum time between two single clicks on the same node, which should start node editing.

**Pascal**

```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 maximum time distance between both clicks in which the edit operation is started.

**See Also**

- Editors and editing

**Class**

- TBaseVirtualTree Class

**Links**

- TBaseVirtualTree Class, See Also

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.EditLink Property**

**TBaseVirtualTree Class**

Keeps a reference to the internal edit link during a node edit operation.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? [Send feedback]*
**TBaseVirtualTree.Expanded Property**

**TBaseVirtualTree Class**

Read or set the expanded state of a particular node.

**Pascal**

```pascal
property Expanded [Node: PVirtualNode]: Boolean;
```

**Description**

Using this property you can expand or collapse the given node. This method uses the central `ToggleNode` method.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.FocusedColumn Property
TBaseVirtualTree Class | See Also

Read or set the currently focused column.

Pascal

```
property FocusedColumn: TColumnIndex;
```

Description

When toExtendedFocus in TVTSelectionOptions 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 Also

FocusedNode, TVTSelectionOptions

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
TBaseVirtualTree.FocusedNode Property

TBaseVirtualTree Class | See Also

Read or set the currently focused node.

Pascal

```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 property is taken into account. Read there for more info about column focus.

See Also

FocusedColumn, TVTSelectionOptions

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
What do you think about this topic? Send feedback!
**TBaseVirtualTree.Font Property**

*TBaseVirtualTree Class*

Same as TWinControl.Font.

**Pascal**

```pascal
property Font;
```

**Description**

See TWinControl.Font.

**Class**

*TBaseVirtualTree Class*

**Links**

*TBaseVirtualTree Class*

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.FullyVisible Property

Read or set whether a node is fully visible or not.

Pascal

```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, which hides the node regardless 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 property, while the expansion state of parents nodes can be examined via the VisiblePath property. If both are true then the node is said to be fully visible.

**See Also**

isVisible, VisiblePath, vsVisible, TVirtualNodeStates

**Class**

TBaseVirtualTree Class

**Links**
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.HasChildren Property

TBaseVirtualTree Class | See Also

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
Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
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

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.HeaderRect Property

TBaseVirtualTree Class

Returns the non-client-area rectangle used for the header.

Pascal

```pascal
property HeaderRect: TRect;
```

Description

Use this property to determine the extents used by the header of Virtual Treeview.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.HintAnimation Property

TBaseVirtualTree Class

Read or set the current hint animation type.

Pascal

```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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.HintMode Property

TBaseVirtualTree Class

Read or set what type of hint you want for the tree view.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!


TBaseVirtualTree.HotCursor Property

TBaseVirtualTree Class | See Also

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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
What do you think about this topic? Send feedback!
**TBaseVirtualTree.HotNode Property**

*TBaseVirtualTree Class | See Also*

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, toHotTrack, TVTPaintOptions*

**Class**

*TBaseVirtualTree Class*

**Links**

*TBaseVirtualTree Class, See Also*

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.Images Property

TBaseVirtualTree Class | See Also

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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
Read or set the current incremental search mode.

Pascal

```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**

- TBaseVirtualTree Class

**Links**

- TBaseVirtualTree Class, See Also
**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 Also**

- IncrementalSearch, IncrementalSearchStart,
- IncrementalSearchTime123out

**Class**

- TBaseVirtualTree Class

**Links**

- TBaseVirtualTree Class, See Also
**TBaseVirtualTree.IncrementalSearchStart Property**

Read or set where to start incremental search.

**Pascal**

```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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also
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, IncrementalSearchDirection, IncrementalSearchStart

Class

TBaseVirtualTree Class
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.Indent Property

TBaseVirtualTree Class

Read or set the indentation amount for node levels.

Pascal

```pascal
property Indent: Cardinal;
```

Description

Each new level in the tree (child nodes of a parent node) are visually shifted to distinguish between them and their parent node (that's the tree layout after all). The Indent property determines the shift distance in pixels.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.IsDisabled Property

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.IsVisible Property

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.LastClickPos Property**

*TBaseVirtualTree Class*

Used for retained drag start and wheel mouse scrolling.

**Pascal**

```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*

**Links**

*TBaseVirtualTree Class*

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.LastDropMode Property

Read how the last drop operation finished.

Pascal

```pascal
property LastDropMode: TDropMode;
```

Description

In the case you don't handle drag'n drop operations directly in OnDragDrop it might be necessary to know how the last drag operation finished. Read more in the drag mode enumeration about what is possible.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.LineMode Property

TBaseVirtualTree Class

Read or set the mode of the tree lines.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.LineStyle Property

TBaseVirtualTree Class

Read or set the mode of the tree lines.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.Margin Property

Read or set the tree's node margin.

Pascal

```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

TBaseVirtualTree Class

Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.MultiLine Property

TBaseVirtualTree Class

Read or toggle the multiline feature for a given node.

Pascal

```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 explicitly specify where to start new lines. The node's height is not automatically adjusted to the given text. Instead there is an event (**OnMeasureItem**), 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

Links

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.NodeAlignment Property

TBaseVirtualTree Class | See Also

Read or set the node alignment value.

Pascal

```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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
TBaseVirtualTree.NodeDataSize Property

TBaseVirtualTree Class | See Also

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

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
TBaseVirtualTree.NodeHeight Property

TBaseVirtualTree Class

Read or set a node's height.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.NodeParent Property
TBaseVirtualTree Class | See Also

Read or set a node's parent node.

Pascal

```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, CopyTo

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.OffsetXY Property

TBaseVirtualTree Class

Read or set the tree's current horizontal and vertical scroll offsets.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnAdvancedHeaderDraw Event
TBaseVirtualTree Class | See Also

Header paint support event.

Pascal

```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
TBaseVirtualTree Class

Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnAfterCellPaint Event

Paint support event.

Pascal

```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

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnAfterItemErase Event**

**TBaseVirtualTree Class | See Also**

Paint support event.

**Pascal**

```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

**Class**

- TBaseVirtualTree Class

**Links**

- TBaseVirtualTree Class, See Also

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.OnAfterItemPaint Event

Paint support event.

Pascal

```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

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
**TBaseVirtualTree.OnAfterPaint Event**

*Paint support event.*

**Pascal**

```plaintext
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

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also

*What do you think about this topic?* Send feedback!
**TBaseVirtualTree.OnBeforeCellPaint Event**

**TBaseVirtualTree Class | See Also**

Paint support event.

**Pascal**

```pascal
property OnBeforeCellPaint: TVTBeforeCellPaintEvent;
```

**Description**

This event is called immediately before a cell is painted.

**See Also**

Paint cycles and stages

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also

---

*What do you think about this topic? Send feedback!*
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

Class

- TBaseVirtualTree Class

Links

- TBaseVirtualTree Class, See Also
TBaseVirtualTree.OnBeforeItemPaint Event

Paint support event.

**Pascal**

```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

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.OnBeforePaint Event

Paint support event.

Pascal

```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

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnChange Event**

**TBaseVirtualTree Class**

Navigation support event.

**Pascal**

```pascal
propertyOnChange:TVTChangeEvent;
```

**Description**

Called when a node's selection state has changed.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.OnChecked Event

TBaseVirtualTree Class

Check support event.

Pascal

```pascal
property OnChecked: TVTChangeEvent;
```

Description

Triggered when a node's check state has changed.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnChecking Event

TBaseVirtualTree Class

Check support event.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnCollapsed Event
TBaseVirtualTree Class

Miscellaneous event.

Pascal

```pascal
property OnCollapsed: TVTChangeEvent;
```

Description
Triggered after a node has been collapsed, that is, its child nodes are no longer displayed.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnCollapsing Event

TBaseVirtualTree Class

Miscellaneous event.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnColumnClick Event
TBaseVirtualTree Class | See Also

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
TBaseVirtualTree Class

Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnColumnDblClick Event**

**TBaseVirtualTree Class | See Also**

Header and column support event.

**Pascal**

```pascal
property OnColumnDblClick: TVTColumnDblClickEvent;
```

**Description**

Same as **OnColumnClick** but for double clicks.

**See Also**

- **OnColumnClick**, **OnHeaderDblClick**

**Class**

- **TBaseVirtualTree Class**

**Links**

- **TBaseVirtualTree Class, See Also**

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.OnColumnResize Event

TBaseVirtualTree Class

Header and column support routine.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnCompareNodes Event

Sort and search support event.

Pascal

```pascal
property OnCompareNodes: TVTCompareEvent;
```

Description

This event is the core event for all comparisons 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 are 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 comparison result as would there be an ascending sort order.

Below is some sample code taken from the Advanced Demo:
procedure TMainForm.VDT1CompareNodes(Sender: TBaseVirtualTree;
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
  TBaseVirtualTree Class

Links
  TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnCreateDataObject Event

**TBaseVirtualTree Class**

Drag'n drop support event.

**Pascal**

```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. Descendants (which override DoGetDataObject) or the application can return an own IDataObject implementation to support special formats.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.OnCreateDragManager Event**

**TBaseVirtualTree Class**

Drag'n drop support event.

**Pascal**

```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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.OnCreateEditor Event

Editing support event.

Pascal

```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. Descendants like TCustomVirtualStringTree supply a generic and simple string editor.

See Also

Editors and editing

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
TBaseVirtualTree.OnDragAllowed Event

Drag'n drop support event.

Pascal

```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 preferred over doing a manual `BeginDrag`.

Notes

The OnDragAllowed event is called only if the current DragMode is `dmManual`.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnDragDrop Event**

**TBaseVirtualTree Class**

Drag'n drop support event.

**Pascal**

```pascal
property OnDragDrop: TVTDragDropEvent;
```

**Description**

Triggered when either a VCL or a OLE drop action occurred. 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.

```pascal
procedure TMainForm.VTDragDrop(Sender: TBaseVirtualTree;
const Formats: array of Word;
Shift: TShiftState;
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 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)
                        AttachMode := amInsertBefore
                    else
                        AttachMode := amNowhere;
            end;
        // in the case the drop target does an optimized move
        // to indicate this also to the drag source
        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;
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnDragOver Event

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
TBaseVirtualTree Class

Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnEditCancelled Event**

**TBaseVirtualTree Class | See Also**

Editing support event.

**Pascal**

```pascal
property OnEditCancelled: TVTEditCancelEvent;
```

**Description**

Triggered when an edit action has been cancelled.

**See Also**

Editors and editing

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also
TBaseVirtualTree.OnEdited Event

TBaseVirtualTree Class | See Also

Editing support event.

Pascal

```pascal
property OnEdited: TVTEditChangeEvent;
```

Description

Triggered when an edit action has successfully been finished.

See Also

Editors and editing

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnEditing Event

TBaseVirtualTree Class | See Also

Editing support event.

Pascal

```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

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnExpanded Event
TBaseVirtualTree Class

Missellaneous event.

Pascal

```
property OnExpanded: TVTChangeEvent;
```

Description

Triggered after a node has been expanded.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree OnExpanding Event

TBaseVirtualTree Class

Miscellaneous event.

Pascal

```pascal
property OnExpanding: TVTChangingEvent;
```

Description

Triggered just before a node is expanded. Use Allowed to allow or deny this action.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnFocusChanged Event

TBaseVirtualTree Class

Navigation support event.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnFocusChanging Event**

**TBaseVirtualTree Class**

Navigation support event.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.OnFreeNode Event**

*TBaseVirtualTree Class*

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*

**Links**

*TBaseVirtualTree Class*

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.OnGetCellIsEmpty Event**

**TBaseVirtualTree Class**

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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class
What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnGetCursor Event**

**TBaseVirtualTree Class**

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

**Links**

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnGetHeaderCursor Event**

**TBaseVirtualTree Class**

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

**Links**

TBaseVirtualTree Class

---

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnGetHelpContext Event

TBaseVirtualTree Class

Miscellaneous event.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnGetImageIndex

Event
TBaseVirtualTree Class

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 they 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

**Links**
TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree::OnGetImageIndexEx Event

TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnGetLineStyle Event

TBaseVirtualTree Class | See Also

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
  TBaseVirtualTree Class

Links
  TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnGetNodeDataSize Event

TBaseVirtualTree Class | See Also

Data management event.

Pascal

```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

Class

TBaseVirtualTree Class
**TBaseVirtualTree.OnGetPopupMenu Event**

**TBaseVirtualTree Class**

Miscellaneous event.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.OnGetUserClipboardFormats Event**

**TBaseVirtualTree Class**

Drag'n drop and clipboard support event.

**Pascal**

```pascal
property OnGetUserClipboardFormats: TVTGetUserClipboardFormats;
```

**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 descendants (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**

TBaseVirtualTree Class
TBaseVirtualTree.OnHeaderClick Event

TBaseVirtualTree Class | See Also

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

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnHeaderDblClick Event**

**TBaseVirtualTree Class | See Also**

Header & column support event.

**Pascal**

```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**

**TBaseVirtualTree Class**

**Links**

**TBaseVirtualTree Class, See Also**

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.OnHeaderDragged Event

TBaseVirtualTree Class

Header & column support event.

Pascal

```pascal
property OnHeaderDragged: TVTHheaderDraggedEvent;
```

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

Links

TBaseVirtualTree Class

---

*What do you think about this topic? [Send feedback]*
**TBaseVirtualTree.OnHeaderDraggedOut Event**

**TBaseVirtualTree Class**

Header & column support event.

**Pascal**

```pascal
property OnHeaderDraggedOut: TVTHeaderDraggedOutEven;
```

**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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.OnHeaderDragging Event**

**TBaseVirtualTree Class**

Header & column support event.

**Pascal**

```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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.HeaderViewEvent

TBaseVirtualTree Class

Header & column support event.

Pascal

```
property OnHeaderView: 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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnHeaderDrawQueryElements Event

Pascal

```pascal
property OnHeaderDrawQueryElements: TVTHeaderPaintQueryElements;
```

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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnHeaderMouseDown Event

TBaseVirtualTree Class

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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnHeaderMouseMove Event

TBaseVirtualTree Class

Header & column support event.

Pascal

```pascal
property OnHeaderMouseMove: TVTHeaderMouseMoveEvent;
```

Description

This event is triggered when the mouse pointer is moved over the header area.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnHeaderMouseUp

Event

TBaseVirtualTree Class

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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnHotChange Event

Navigation support event.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree class:

TBaseVirtualTree.OnIncrementalSearch event:

Miscellaneous event.

Pascal:

```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.

```pascal
procedure TfrmProperties.VST3IncrementalSearch(Sender: TObject; var Result: Integer);
var
  S, PropText: string;
begin
  // Note: This code requires a proper Unicode/Wide
```
// size and clarity reasons. For now strings are
// 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
// 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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnInitChildren Event

Node management event.

Pascal

```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 initialization without 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

Class

- TBaseVirtualTree Class

Links

- TBaseVirtualTree Class, See Also
What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnInitNode Event**

**TBaseVirtualTree Class | See Also**

Node management event.

**Pascal**

```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

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.OnKeyAction Event

TBaseVirtualTree Class

Miscellaneous event.

Pascal

```pascal
property OnKeyAction: TVTKeyActionEvent;
```

Description

This event is a convenient way for the application or descendant 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnLoadNode Event

Streaming support event.

Pascal

```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

`TBaseVirtualTree Class`

Links

`TBaseVirtualTree Class`, `See Also`
What do you think about this topic? Send feedback!
TBaseVirtualTree.OnMeasureItem Event

TBaseVirtualTree Class | See Also

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

    TBaseVirtualTree Class
Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnNodeCopied Event

TBaseVirtualTree Class

Miscellaneous event.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnNodeCopying Event

TBaseVirtualTree Class

Miscellaneous event.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnNodeMoved Event

TBaseVirtualTree Class

Miscellaneous event.

Pascal

```pascal
property OnNodeMoved: TVTNodeMovedEvent;
```

Description

This event is very much like OnNodeCopied but used for moving nodes instead.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnNodeMoving Event

TBaseVirtualTree Class

Miscellaneous event.

Pascal

```
property OnNodeMoving: TVTNodeMovingEvent;
```

Description

This event is very much like OnNodeCopying but used for moving nodes instead.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree. OnPaintBackgroundColor Event

TBaseVirtualTree Class

Paint support event.

Pascal

property OnPaintBackgroundColor: 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnRenderOLEData Event**

**TBaseVirtualTree Class**

Drag'n drop and clipboard support event.

**Pascal**

```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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.OnResetNode Event

TBaseVirtualTree Class | See Also

Node management event.

Pascal

```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
TBaseVirtualTree Class

Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnSaveNode Event
TBaseVirtualTree Class | See Also

Streaming support event.

Pascal

```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
TBaseVirtualTree Class

Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnScroll Event

TBaseVirtualTree Class | See Also

Miscellaneous event.

Pascal

```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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree Class

TBaseVirtualTree.OnShowScrollbar Event

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OnStateChange Event**

**TBaseVirtualTree Class**

Miscellaneous event.

**Pascal**

```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

**Links**

TBaseVirtualTree Class
**TBaseVirtualTree.OnStructureChange Event**

**TBaseVirtualTree Class**

Miscellaneous event.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.OnUpdating Event

TBaseVirtualTree Class

Miscellaneous event.

Pascal

```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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.RootNode Property

Reference to the internal root node which is the anchor of the entire tree node hierarchy.

Pascal

```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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.RootNodeCount Property**

Read or set the number of nodes on the top level.

**Pascal**

```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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.ScrollBarOptions Property**

TBaseVirtualTree Class

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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.SearchBuffer Property**

**TBaseVirtualTree Class | See Also**

Current input string for incremental search.

**Pascal**

```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

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.Selected Property

TBaseVirtualTree Class

Property to modify or determine the selection state of a node.

Pascal

```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. 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.SelectedCount Property

TBaseVirtualTree Class

Contains the number of selected nodes.

Pascal

```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` or `AddToSelection/RemoveFromSelection`.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.SelectionBlendFactor

Property

TBaseVirtualTree Class | See Also

Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

Pascal

```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
TBaseVirtualTree Class

Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.SelectionCurveRadius Property

Read or set the current corner radius for node selection rectangles.

Pascal

```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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
**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**). Another one is the state image and yet another one the normal/selected image.

**See Also**

CheckImages, Images

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also
**TBaseVirtualTree.TextMargin Property**

Read or set the distance of the node caption to its borders.

**Pascal**

```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**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also
What do you think about this topic? Send feedback!
**TBaseVirtualTree.TopNode Property**

**TBaseVirtualTree Class**

The top node is the node which is currently at the top border of the client area.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.TotalCount Property**

*TBaseVirtualTree Class*

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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.TotalInternalDataSize Property**

Keeps the currently accumulated data size for one node.

**Pascal**

```pascal
property TotalInternalDataSize: Cardinal;
```

**Description**

Each node in the tree not only supports user data but also an internal area where TVirtualBaseTree descendants 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

**Class**

- TBaseVirtualTree Class

**Links**
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
**TBaseVirtualTree.TreeOptions** Property

**TBaseVirtualTree Class**

Reference to the tree's options.

**Pascal**

```pascal
property TreeOptions: TCustomVirtualTreeOptions;
```

**Description**

The tree options are one of the main switches 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**

**TBaseVirtualTree Class**

**Links**

**TBaseVirtualTree Class**

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.TreeStates Property**

TBaseVirtualTree Class | See Also

Property which keeps a set of flags which indicate current operation and states of the tree.

**Pascal**

```pascal
property TreeStates: TVirtualTreeStates;
```

**Description**

Often it is extremely 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

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also
What do you think about this topic? Send feedback!
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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.VerticalAlignment Property**

TBaseVirtualTree Class

Used to set a node's vertical button alignment 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`. 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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.VisibleCount Property**

*Class*

TBaseVirtualTree Class

**Description**

Number of currently visible nodes.

Visible nodes are those nodes which have the vsVisible flag set in their states.

**Pascal**

```pascal
property VisibleCount: Cardinal;
```

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.VisiblePath Property

TBaseVirtualTree Class | See Also

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

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.WantTabs Property

TBaseVirtualTree Class

Read or set whether the tree wants to process tabs on its own.

Pascal

```pascal
property WantTabs: Boolean;
```

Description

Usually tab key 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

Links

TBaseVirtualTree Class
TBaseVirtualTree.AbsoluteIndex Method

TBaseVirtualTree Class

Reads the overall index of a node.

Pascal

```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 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

Links

TBaseVirtualTree Class
What do you think about this topic? Send feedback!
TBaseVirtualTree.AddChild Method

TBaseVirtualTree Class  |  See Also

Creates and adds a new child node to given node.

Pascal

```pascal
function AddChild(Parent: PVirtualNode; UserData: Pointer = nil): Integer;
```

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 utilizing its virtual paradigm. Important advantages will so disappear. If possible you should restructure your design and try to use the right way: via `OnInitNode` and `OnInitChildren`.

See Also

`InsertNode`, `OnInitNode`, `OnInitChildren`
Class
  TBaseVirtualTree Class

Links
  TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.AddFromStream Method

TBaseVirtualTree Class | See Also

Adds the content from the given stream to the given node.

Pascal

```
procedure AddFromStream(Stream: TStream; TargetNode:
```

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 Also

SaveToStream

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
TBaseVirtualTree.AddToSelection Method (PVirtualNode)

TBaseVirtualTree Class

Adds one or more nodes to the current selection.

Pascal

procedure AddToSelection(Node: PVirtualNode); virtual

Description

AddToSelection either takes a single node or an array of nodes and adds them to the current selection 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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.AddToSelection Method (TNodeArray, Integer, Boolean)

```pascal
procedure AddToSelection(const NewItems: TNodeArray;
procedure AddToSelection(const NewItems: TNodeArray;
```

Class
- TBaseVirtualTree Class

Links
- TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.AdjustPaintCellRect Method

Used in descendents to modify the clip rectangle of the current column while painting a certain node.

Pascal

```
procedure AdjustPaintCellRect(var PaintInfo: TVTPaint);
```

Description

The rectangle for the given cell (node, column pair in PaintInfo) can be adjusted by descendents to make room for special drawings, if necessary.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.AdjustPanningCursor Method**

**TBaseVirtualTree Class**

Loads the proper cursor which indicates into which direction scrolling is done.

**Pascal**

```pascal
procedure AdjustPanningCursor(X: Integer; Y: Integer);
```

**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 *wheel 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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.AdviseChangeEvent Method
TBaseVirtualTree Class

Used to register a delayed change event.

Pascal

```pascal
procedure AdviseChangeEvent(StructureChange: Boolean)
```

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) or, if BeginUpdate was called, until EndUpdate is executed. If StructureChange 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 to 0, the other one is to enter the synchronous mode by calling BeginSynch.

Class

TBaseVirtualTree Class
Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.AllocateInternalDataArea Method**

Registration method to allocate tree internal data per node.

**Pascal**

```pascal
function AllocateInternalDataArea(Size: Cardinal): Cardinal;
```

**Description**

This method is used for descendants 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` (as shown in `TCustomVirtualStringTree`) which does the pointer mathematic.
See Also
- Data handling, TotalInternalDataSize

Class
- TBaseVirtualTree Class

Links
- TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.Animate Method

TBaseVirtualTree Class

Support method for animated actions in the tree view.

Pascal

```
procedure Animate(Steps: Cardinal; Duration: Cardinal; Callback: Description);
```

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.Assign Method

TBaseVirtualTree Class

Used to copy properties from another Virtual Treeview.

Pascal

```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) for the header assignment. In order to copy the nodes you must save them to a stream and restore them in the other control.
TBaseVirtualTree_BeginDrag Method

TBaseVirtualTree Class

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 Treeview. If you selectively want to allow to start a drag operation then use the OnDragAllowed event which is called when DragMode is dmManual.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.BeginSynch Method**

**TBaseVirtualTree Class**

Enters the tree into a special synchronized mode.

**Pascal**

```pascal
procedure BeginSynch;
```

**Description**

Similar to **BeginUpdate** 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**).

**Class**

**TBaseVirtualTree Class**

**Links**

**TBaseVirtualTree Class**

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.BeginUpdate Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.CalculateSelectionRect Method

Support method for draw selection.

Pascal

```pascal
function CalculateSelectionRect(X: Integer; Y: Integer): Boolean;
```

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.CanAutoScroll Method

TBaseVirtualTree Class

Determines whether the tree can currently auto scroll its window.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.CancelCutOrCopy Method

TBaseVirtualTree Class

Cancels any pending cut or copy clipboard operation.

Pascal

procedure CancelCutOrCopy;

Description

This method is used to stop any pending clipboard operation. No data is transferred nor are nodes deleted.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
TBaseVirtualTree.CancelEditNode Method

TBaseVirtualTree Class

Cancel the current edit operation, if there is any.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.CanEdit Method

TBaseVirtualTree Class

Determines whether a node can be edited or not.

Pascal

function CanEdit(Node: PVirtualNode; Column: TColumn):

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.CanFocus Method**

**TBaseVirtualTree Class**

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

**Links**

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.CanShowDragImage Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.Change Method

Central method called when a node's selection state changes.

Pascal

```pascal
procedure Change(Node: PVirtualNode); virtual;
```

Description

The Change method is called to trigger the change notification chain. Depending on the sync and the update states of the tree as well as the ChangeDelay 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, EndSynch, BeginUpdate, EndUpdate, ChangeDelay

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
TBaseVirtualTree.ChangeScale Method

TBaseVirtualTree Class | See Also

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, DefaultNodeHeight

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.CheckParentCheckState Method

Helper method for recursive check state changes.

Pascal

```
function CheckParentCheckState(Node: PVirtualNode; NewCheckState: TCheckState);
```

Description

Checks all siblings of node to determine which check state Node's parent must get.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.Clear Method**

**TBaseVirtualTree Class**

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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.ClearChecked Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure ClearChecked;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.ClearSelection Method

TBaseVirtualTree Class

Removes all nodes from the current selection.

Pascal

```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

Links

TBaseVirtualTree Class
**TBaseVirtualTree.ClearTempCache Method**

**TBaseVirtualTree Class**

Helper method to clear the internal temporary node cache.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree(ColumnIsEmpty) Method

TBaseVirtualTree Class | See Also

Used to determine if a cell is considered as being empty.

Pascal

```
function ColumnIsEmpty(Node: PVirtualNode; Column: TColumnIndex) ;
```

Description

An empty cell might be used for the automatic column spanning feature. Descendants can override this method to modify the tree's behavior.

See Also

toAutoSpanColumns

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also
TBaseVirtualTree.CopyTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean)

Copies Source and all its child nodes to Target.

Pascal

```pascal
function CopyTo(Source: PVirtualNode; Tree: TBaseVirtualTree; Target: PVirtualNode; Mode: TVTNodeAttachMode; ChildrenOnly: Boolean): PVirtualNode;
```

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
What do you think about this topic? Send feedback!
### TBaseVirtualTree.CopyToClipBoard Method

**TBaseVirtualTree Class**

Copies all currently selected nodes to the clipboard.

**Pascal**

```pascal
procedure CopyToClipBoard; virtual;
```

**Description**

CopyToClipboard causes the tree to copy the currently selected nodes to the clipboard. Actually, Virtual Treeview maintains so-called delayed rendering. This means the participating nodes are marked as being in the current clipboard set (see vsCutOrCopy in TVirtualNodeStates) 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.

**Class**

TBaseVirtualTree Class
TBaseVirtualTree.CountLevelDifference Method

TBaseVirtualTree Class

Determines the level difference of two nodes.

Pascal

```pascal
function CountLevelDifference(Node1: PVirtualNode; Node2: PVirtualNode): Integer;
```

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(Node2) - GetNodeLevel(Node1)`, but with sign. If the result is negative then Node2 is less intended than Node1.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.CountVisibleChildren Method

TBaseVirtualTree Class

Determines the number of visible child nodes of the given node.

Pascal

```pascal
function CountVisibleChildren(Node: PVirtualNode): Cardinal;
```

Description

CountVisibleChildren iterates through all child nodes of `Node` and counts how many of them have the `vsVisible` state set.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
**TBaseVirtualTree.Create Constructor**

**TBaseVirtualTree Class**

Constructor of the control

**Pascal**

```
constructor Create(AOwner: TComponent); override;
```

**Description**

The constructor initializes certain properties to their default values.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.CreateParams Method**

**TBaseVirtualTree Class**

Prepares the creation of the controls window handle.

**Pascal**

```
procedure CreateParams(var Params: TCreateParams);
```

**Description**

CreateParams is overridden to allow to set certain window styles for the control.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.CreateWnd Method**

**TBaseVirtualTree Class**

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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic?*  [Send feedback!](#)
**TBaseVirtualTree.CutToClipBoard Method**

**TBaseVirtualTree Class**

Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

**Pascal**

```pascal
procedure CutToClipBoard; virtual;
```

**Description**

Similar to `CopyToClipboard` only the nodes are deleted after they have been pasted into the target.

**Class**

`TBaseVirtualTree Class`

**Links**

`TBaseVirtualTree Class`

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DefineProperties Method

TBaseVirtualTree Class

Helper method to customize loading and saving persistent tree data.

Pascal

```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

Links

TBaseVirtualTree Class
What do you think about this topic? Send feedback!
**TBaseVirtualTree.DeleteChildren Method**

**TBaseVirtualTree Class**

Removes all child nodes from the given node.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

---

What do you think about this topic? Send feedback!
TBaseVirtualTree.DeleteNode Method

Removes the given node from the tree.

Pascal

```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 or InsertNode then the event OnFreeNode is called to allow the application to free any user data attached to a node.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
**TBaseVirtualTree.DeleteSelectedNodes Method**

**TBaseVirtualTree Class**

Removes all currently selected nodes from the tree.

**Pascal**

```
procedure DeleteSelectedNodes; virtual;
```

**Description**

All nodes in the current selection are affected.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.Destroy Destructor**

**TBaseVirtualTree Class**

Destructor of the control.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DetermineHiddenChildrenFlag Method**

**TBaseVirtualTree Class** | **See Also**

Determines whether all children of a given node are hidden.

**Pascal**

```pascal
procedure DetermineHiddenChildrenFlag(Node: PVirtual)
```

**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

**Links**

TBaseVirtualTree Class, See Also

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DetermineHiddenChildrenFlagAllNodes Method

**Description**

As extension to DetermineHiddenChildren 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.

**Pascal**

```pascal
procedure DetermineHiddenChildrenFlagAllNodes; virtual
```

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DetermineHitPositionLTR Method**

**TBaseVirtualTree Class**

Determines the hit position within a node with left-to-right and right-to-left orientation.

**Pascal**

```pascal
procedure DetermineHitPositionLTR(var HitInfo: THitInfo);
procedure DetermineHitPositionRTL(var HitInfo: THitInfo);
```

**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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DetermineNextCheckState Method

TBaseVirtualTree Class

Not documented.

Pascal

```
function DetermineNextCheckState(CheckType: TCheckType): Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DetermineScrollDirections Method

TBaseVirtualTree Class

Not documented.

Pascal

function DetermineScrollDirections(X: Integer; Y: Integer): TBaseVirtualTree;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoAdvancedHeaderDraw Method**

*TBaseVirtualTree Class*

Not documented.

**Pascal**

```pascal
procedure DoAdvancedHeaderDraw(var PaintInfo: THeaderPaintInfo)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

*TBaseVirtualTree Class*

**Links**

*TBaseVirtualTree Class*

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoAfterCellPaint Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure DoAfterCellPaint(Canvas: TCanvas; Node: PV);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoAfterItemErase Method

TBaseVirtualTree Class

Not documented.

Pascal

```
procedure DoAfterItemErase(Canvas: TCanvas; Node: PV
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoAfterItemPaint Method
TBaseVirtualTree Class

Not documented.

Pascal

procedure DoAfterItemPaint(Canvas: TCanvas; Node: PV;)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoAfterPaint Method
TBaseVirtualTree Class

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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree DoAutoScroll Method

TBaseVirtualTree Class

Enables or disables the auto scroll timer.

Pascal

```
procedure DoAutoScroll(X: Integer; Y: Integer);
```

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 method.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoBeforeCellPaint Method

TBaseVirtualTree Class

Not documented.

Pascal

```
procedure DoBeforeCellPaint(Canvas: TCanvas; Node: P
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoBeforeDrag Method
TBaseVirtualTree Class

Not documented.

Pascal

```
function DoBeforeDrag(Node: PVirtualNode; Column: TColumnIndex):
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoBeforeItemErase Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure DoBeforeItemErase(Canvas: TCanvas; Node: P

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoBeforeItemPaint Method**

TBaseVirtualTree Class

Not documented.

**Pascal**

```pascal
function DoBeforeItemPaint(Canvas: TCanvas; Node: PV)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic?* Send feedback!
TBaseVirtualTree.DoBeforePaint Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoCancelEdit Method**

**TBaseVirtualTree Class**

Called when the tree should stop editing without accepting changed values.

**Pascal**

```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** is triggered and a message is sent to release the edit link asynchronously.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoCanEdit Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure DoCanEdit(Node: PVirtualNode; Column: TCol);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoChange Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class
**TBaseVirtualTree.DoCheckClick Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
procedure DoCheckClick(Node: PVirtualNode; NewCheckS
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DoChecked Method**

**TBaseVirtualTree Class**

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

**Links**

TBaseVirtualTree Class

*What do you think about this topic?* Send feedback!
TBaseVirtualTree.DoChecking Method

TBaseVirtualTree Class

Not documented.

Pascal

```
function DoChecking(Node: PVirtualNode; var NewCheckState: Description);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoCollapsed Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoCollapsing Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
function DoCollapsing(Node: PVirtualNode): Boolean;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class
**TBaseVirtualTree.DoColumnClick Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
procedure DoColumnClick(Column: TTreeNode; Shift: TShiftState);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TBaseVirtualTree Class**

**Links**

**TBaseVirtualTree Class**

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DoColumnDblClick Method**

*TBaseVirtualTree Class*

Not documented.

**Pascal**

```pascal
procedure DoColumnDblClick(Column: TColumnIndex; Shift: TShiftState);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

*TBaseVirtualTree Class*

**Links**

*TBaseVirtualTree Class*

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DoColumnResize Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoCompare Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
function DoCompare(Node1: PVirtualNode; Node2: PVirtualNode);`n```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoCreateDataObject Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoCreateDragManager Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoCreateEditor Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
function DoCreateEditor(Node: PVirtualNode; Column: ...
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoDragDrop Method
TBaseVirtualTree Class

Not documented.

Pascal

procedure DoDragDrop(Source: TObject; DataObject: IDataObject; Formats: TFormatID);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoDragExpand Method
TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoDragging Method
TBaseVirtualTree Class

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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoDragOver Method
TBaseVirtualTree Class

Not documented.

Pascal

function DoDragOver(Source: TObject; Shift: TShiftState; State: TDragState; Pt: TPoint; Mode: Integer): Boolean;

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoEdit Method

TBaseVirtualTree Class | See Also

Initiates editing of the currently set focused column and edit node.

Pascal

```pascal
procedure DoEdit; virtual;
```

Description

This method takes care for editor creation and initialization. You can look for tsEditing in TreeStates to know whether editing is currently active.

See Also

tsEditing, OnCreateEditor, IVTEditLink

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoEndDrag Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure DoEndDrag(Target: TObject; X: Integer; Y: Integer);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoEndEdit Method**

**TBaseVirtualTree Class | See Also**

Stops the current edit operation and takes over the new content.

**Pascal**

```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 overrides this method to tell the application about the new caption by calling OnNewText.

**See Also**

DoEdit, OnNewText, EditNode

**Class**

TBaseVirtualTree Class
Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoExpanded	Method

TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoExpanding Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoFocusChange Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure DoFocusChange(Node: PVirtualNode; Column:
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoFocusChanging**

**Method**

TBaseVirtualTree Class

Not documented.

**Pascal**

```pascal
function DoFocusChanging(OldNode: PVirtualNode; NewNode: PVirtualNode): integer;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoFocusNode Method

TBaseVirtualTree Class

Internal method to set the focused node.

Pascal

```pascal
procedure DoFocusNode(Node: PVirtualNode; Ask: Boolean);
```

Description

This method 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` event.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoFreeNode Method
TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoGetAnimationType Method**

_TBaseVirtualTree Class_

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 switched off entirely. Also this case is handled by this method.

**Class**

_TBaseVirtualTree Class_

**Links**

_TBaseVirtualTree Class_
What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoGetCursor Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```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**

**Links**

**TBaseVirtualTree Class**

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DoGetHeaderCursor**

**Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```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**

**Links**

**TBaseVirtualTree Class**

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoGetImageIndex Method

Pascal

function DoGetImageIndex(Node: PVirtualNode; Kind: T

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
**TBaseVirtualTree.DoGetLineStyle Method**

**Class**

TBaseVirtualTree Class

Not documented.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoGetNodeHint Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
function DoGetNodeHint(Node: PVirtualNode; Column: TColumnIndex): ResultType;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoGetNodeTooltip Method
TBaseVirtualTree Class

Not documented.

Pascal

function DoGetNodeTooltip(Node: PVirtualNode; Column

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoGetNodeWidth Method**

**TBaseVirtualTree Class**

Overridable method which always returns 0.

**Pascal**

```pascal
function DoGetNodeWidth(Node: PVirtualNode; Column:
```

**Description**

Descendants 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**

`TBaseVirtualTree Class`

**Links**

`TBaseVirtualTree Class`

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DoGetPopupMenu Method**

**Class**

TBaseVirtualTree Class

**Description**

This method does an automatic parent traversal in the tree hierarchy to find a matching popup menu.

**Pascal**

```pascal
function DoGetPopupMenu(Node: PVirtualNode; Column:
```

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoGetUserClipboardFormats Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure DoGetUserClipboardFormats(var Formats: TFormatEtcArray);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoHeaderButtonClick Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure DoHeaderButtonClick(Column: TColumnIndex; Button

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoHeaderDb1Click Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
procedure DoHeaderDb1Click(Column: TColumnIndex; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoHeaderDragged
Method
TBaseVirtualTree Class

Not documented.

Pascal

```
procedure DoHeaderDragged(Column: TColumnIndex; OldPosition: Integer);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoHeaderDraggedOut Method

TBaseVirtualTree Class

Not documented.

Pascal

```
procedure DoHeaderDraggedOut(Column: TColumnIndex; DropPosition: TPoint);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoHeaderDragging Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
function DoHeaderDragging(Column: TColumnIndex): Boolean;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoHeaderDraw Method
TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure DoHeaderDraw(Canvas: TCanvas; Column: TVirtualTreeColumn);
```

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoHeaderDrawQueryElements Method

TBaseVirtualTree Class

Not documented.

Pascal

```
procedure DoHeaderDrawQueryElements(var PaintInfo: T
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoHeaderMouseDown Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
procedure DoHeaderMouseDown(Button: TMouseButton; Shift: TShiftState; X: Integer; Y: Integer);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoHeaderMouseMove Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure DoHeaderMouseMove(Shift: TShiftState; X: Integer);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoHeaderMouseUp Method
TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure DoHeaderMouseUp(Button: TMouseButton; Shift: TShiftState; X: Integer; Y: Integer);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoHotChange Method
TBaseVirtualTree Class

Not documented.

Pascal

```
procedure DoHotChange(Old: PVirtualNode; New: PVirtualNode);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoIncrementalSearch Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
function DoIncrementalSearch(Node: PVirtualNode; const Description: String): Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoInitChildren Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure DoInitChildren(Node: PVirtualNode; var ChildCount):

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoInitNode Method**

**Class**

TBaseVirtualTree Class

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Pascal**

```
procedure DoInitNode(Parent: PVirtualNode; Node: PVirtualNode);
```

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoKeyAction Method

TBaseVirtualTree Class

Not documented.

Pascal

function DoKeyAction(var CharCode: Word; var Shift: TShiftState): Boolean;

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoLoadUserData Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure DoLoadUserData(Node: PVirtualNode; Stream:
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoMeasureItem Method

TBaseVirtualTree Class

Not documented.

Pascal

```
procedure DoMeasureItem(TargetCanvas: TCanvas; Node: ...)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoNodeCopied Method

TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoNodeCopying Method

Pascal

function DoNodeCopying(Node: PVirtualNode; NewParent

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoNodeMoved Method

TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoNodeMoving Method

TBaseVirtualTree Class

Not documented.

Pascal

```
function DoNodeMoving(Node: PVirtualNode; NewParent: TBaseVirtualTree);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoPaintBackgroundColor**

**Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
function DoPaintBackgroundColor(Canvas: TCanvas; R: TRect): Boolean;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TBaseVirtualTree Class**

**Links**

**TBaseVirtualTree Class**

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoPaintDropMark Method

Overridable method which draws the small line on top of a nodes image depending on the current drop state.

Pascal

```pascal
procedure DoPaintDropMark(Canvas: TCanvas; Node: PVirtualNode)
```

Description

This method draws a simple polyline using Colors.DropMarkColor. Descendant can override this method to customize the appearance of the drop mark.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoPaintNode Method**

**TBaseVirtualTree Class**

Overridable method which does nothing.

**Pascal**

```
procedure DoPaintNode(var PaintInfo: TVTPaintInfo);
```

**Description**

Descendants override this method to paint the content of the node. For instance string trees draw the node's caption.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DoPopupMenu Method**

**TBaseVirtualTree Class**

Overridable method which shows the popup menu for the given node.

**Pascal**

```pascal
procedure DoPopupMenu(Node: PVirtualNode; Column: TColumnIndex);
```

**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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoRenderOLEData Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
function DoRenderOLEData(const FormatEtcIn: TFormatEtc): Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoReset Method
TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoSaveUserData Method
TBaseVirtualTree Class

Not documented.

Pascal

```
procedure DoSaveUserData(Node: PVirtualNode; Stream: TStream);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoScroll Method**

**TBaseVirtualTree Class**

Overridable method which triggers the **OnScroll** event.

**Pascal**

```pascal
procedure DoScroll(DeltaX: Integer; DeltaY: Integer);
```

**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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoSetOffsetXY Method

TBaseVirtualTree Class

Internal core routine to set the tree's scroll position.

Pascal

```pascal
function DoSetOffsetXY(Value: TPoint; Options: TScrollUpdateOptions)
```

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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoShowScrollbar Method
TBaseVirtualTree Class

Not documented.

Pascal

```
procedure DoShowScrollbar(Bar: Integer; Show: Boolean);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoStartDrag Method

Not documented.

Pascal

```pascal
procedure DoStartDrag(var DragObject: TDragObject);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoStateChange

Method

TBaseVirtualTree Class

Not documented.

Pascal

```
procedure DoStateChange(Enter: TVirtualTreeStates; Leave: TVirtualTreeStates);...
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoStructureChange

Method
TBaseVirtualTree Class

Not documented.

Pascal

procedure DoStructureChange(Node: PVirtualNode; Reason:

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DoTimerScroll Method

Callback method which is triggered whenever the scroll timer fires.

Pascal

```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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DoUpdating Method**

**TBaseVirtualTree Class**

---

Not documented.

**Pascal**

```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**

**Links**

**TBaseVirtualTree Class**

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DoValidateCache Method
TBaseVirtualTree Class

Not documented.

Pascal

```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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.DragCanceled Method**

**TBaseVirtualTree Class**

Called by the VCL when a drag'n drop operation was canceled by the user.

**Pascal**

```pascal
procedure DragCanceled; override;
```

**Description**

DragCanceled is used to do some housekeeping in the tree.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DragDrop Method

TBaseVirtualTree Class

Helper method, which is used when a drag operation is finished.

Pascal

```pascal
function DragDrop(const DataObject: IDataObject; KeyState: Integer; Pt: TPoint): Boolean;
```

Description

This method is called by the TVTDragManager.Drop and prepares the list of available clipboard formats to be passed to DoDragDrop.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DragEnter Method
TBaseVirtualTree Class

Not documented.

Pascal

```
function DragEnter(KeyState: Integer; Pt: TPoint; var Description: string):
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DragFinished Method

TBaseVirtualTree Class

Called when a drag operation is finished (accepted or cancelled).

Pascal

procedure DragFinished; virtual;

Description

This method is internally used to make up for the swallowed mouse-up messages during drag' drop.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.Dragging Method**

**TBaseVirtualTree Class**

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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.DragLeave Method**

*TBaseVirtualTree Class*

Not documented.

**Pascal**

```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*

**Links**

*TBaseVirtualTree Class*

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.DragOver Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
function DragOver(Source: TObject; KeyState: Integer;
DragState: TDragState; Pt: TPoint;
)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DrawDottedHLine Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure DrawDottedHLine(const PaintInfo: TVTPaintInfo);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.DrawDottedVLine Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure DrawDottedVLine(const PaintInfo: TVTPaintInfo);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree/EditNode** Method

**TBaseVirtualTree Class | See Also**

Starts editing the given node if allowed to.

**Pascal**

```pascal
function EditNode(Node: PVirtualNode; Column: TColumn);
```

**Description**

This method can be used by the application to manually start editing 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

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also
What do you think about this topic? Send feedback!
**TBaseVirtualTree.EndEditNode Method**

**TBaseVirtualTree Class | See Also**

Stops node editing if it was started before.

**Pascal**

```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, EditNode, DoEdit

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.EndSynch Method

Counterpart to BeginSynch.

Pascal

```pascal
procedure EndSynch;
```

Description

Counts down the internal synchronous mode counter and ends synchronous mode when this counter reaches zero.

See Also

BeginSynch, BeginUpdate, EndUpdate

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.EndUpdate Method

TBaseVirtualTree Class

Resets the update lock set by BeginUpdate.

Pascal

```
procedure EndUpdate;
```

Description

This method is the counterpart to BeginUpdate 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

Links

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.ExecuteAction Method

TBaseVirtualTree Class

Not documented.

Pascal

```
function ExecuteAction(Action: TBasicAction): Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.FindNodeInSelection Method
TBaseVirtualTree Class

Helper method to find the given node in the current selection.

Pascal

function FindNodeInSelection(P: PVirtualNode; var Index: Integer): Boolean;

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.FinishChunkHeaderView

**TBaseVirtualTree.FinishChunkHeaderView Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
procedure FinishChunkHeader(Stream: TStream; StartPos: Integer; EndPos: Integer);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.FinishCutOrCopy Method**

**TBaseVirtualTree Class**

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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.FontChanged Method
TBaseVirtualTree Class

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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.FullCollapse Method

Collapses all nodes in the tree.

Pascal

```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

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
**TBaseVirtualTree.FullExpand Method**

*TBaseVirtualTree Class*

Expands all nodes in the tree.

**Pascal**

```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.

**Class**

*TBaseVirtualTree Class*

**Links**

*TBaseVirtualTree Class*

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.GetBorderDimensions

**Method**

TBaseVirtualTree Class

- Not documented.

**Pascal**

```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

**Links**

- TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetCheckImage Method

Pascal

```
function GetCheckImage(Node: PVirtualNode): Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetCheckImageListFor Method

TBaseVirtualTree Class

Not documented.

Pascal

```
class function GetCheckImageListFor(Kind: TCheckImageKind): TCheckImageList;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetColumnClass Method

TBaseVirtualTree Class

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 but descendents can override this method to return own classes.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
TBaseVirtualTree.GetControlsAlignment Method
TBaseVirtualTree Class

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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetDisplayRect Method

TBaseVirtualTree Class

Returns the visible region used by the given node in client coordinates.

Pascal

function GetDisplayRect(Node: PVirtualNode; Column: TColumnIndex) returning 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 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.GetFirst Method**

**TBaseVirtualTree Class**

Group of node navigation functions.

**Pascal**

```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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetFirst</td>
<td>First node in the tree with initialization.</td>
</tr>
<tr>
<td>GetFirstChild</td>
<td>First child node with initialization.</td>
</tr>
<tr>
<td>GetFirstCutCopy</td>
<td>First node in cut/copy set (no initialization needed).</td>
</tr>
<tr>
<td>GetFirstInitialized</td>
<td>First initialized node in the tree (no initialization needed).</td>
</tr>
<tr>
<td>GetFirstNoInit</td>
<td>First node in the tree without initialization.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetFirstVisible</td>
<td>First visible node in the tree with initialization.</td>
</tr>
<tr>
<td>GetFirstVisibleChild</td>
<td>First visible child of a node with initialization.</td>
</tr>
<tr>
<td>GetFirstVisibleChildNoInit</td>
<td>First visible child of a node without initialization.</td>
</tr>
<tr>
<td>GetFirstVisibleNoInit</td>
<td>First visible node in the tree without initialization.</td>
</tr>
</tbody>
</table>

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.GetHeaderClass Method**

TBaseVirtualTree Class

Returns the header class to be used by the tree.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetHintWindowClass Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetHitTestInfoAt Method

TBaseVirtualTree Class

Returns information about the node at the given position.

Pascal

```pascal
procedure GetHitTestInfoAt(X: Integer; Y: Integer; Relative: Boolean; HitInfo: THitTestInfo);
```

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.
Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetImageIndex Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure GetImageIndex(var Info: TVTPaintInfo; Kind

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.GetLast Method**

*TBaseVirtualTree Class*

Group of node navigation functions.

**Pascal**

```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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetLast</td>
<td>Last node in the tree with initialization.</td>
</tr>
<tr>
<td>GetLastChild</td>
<td>Last child node with initialization.</td>
</tr>
<tr>
<td>GetLastChildNoInit</td>
<td>Last child node without initialization.</td>
</tr>
<tr>
<td>GetLastInitialized</td>
<td>Last initialized node in the tree (no initialization needed).</td>
</tr>
<tr>
<td>GetLastNoInit</td>
<td>Last node in the tree without initialization.</td>
</tr>
<tr>
<td>GetLastVisibleNoInit</td>
<td>Last visible node in the tree with</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetLastVisible</td>
<td>initialization.</td>
</tr>
<tr>
<td>GetLastVisibleChild</td>
<td>Last visible child of a node with initialization.</td>
</tr>
<tr>
<td>GetLastVisibleChildNoInit</td>
<td>Last visible child of a node without initialization.</td>
</tr>
<tr>
<td>GetLastVisibleNoInit</td>
<td>Last visible node in the tree without initialization.</td>
</tr>
</tbody>
</table>

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.GetMaxColumnWidth Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetMaxRightExtend Method

Determines the maximum width of the currently visible part of the tree.

Pascal

```pascal
function GetMaxRightExtend: Cardinal; virtual;
```

Description

This method is similar to `GetMaxColumnWidth`, 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

Links

TBaseVirtualTree Class
**TBaseVirtualTree.GetNativeClipboardFormats Method**

**TBaseVirtualTree Class**

Used to let descendants and the application add their own supported clipboard formats.

**Pascal**

```pascal
procedure GetNativeClipboardFormats(var Formats: TFormatEtcArray);
```

**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 property as well as any changes made by the OnGetUserClipboardFormats event if a handler for it is attached.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetNext Method

TBaseVirtualTree Class

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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetNext</td>
<td>Next node in the tree with initialization.</td>
</tr>
<tr>
<td>GetNextCutCopy</td>
<td>Next node in the cut/copy set (no initialization needed).</td>
</tr>
<tr>
<td>GetNextInitialized</td>
<td>Next initialized node in the tree (no initialization needed).</td>
</tr>
<tr>
<td>GetNextNoInit</td>
<td>Next node in the tree without initialization.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>GetNextSelected</td>
<td>Next selected node (no initialization needed).</td>
</tr>
<tr>
<td>GetNextSibling</td>
<td>Next sibling node with initialization.</td>
</tr>
<tr>
<td>GetNextVisible</td>
<td>Next visible node in the tree with initialization.</td>
</tr>
<tr>
<td>GetNextVisibleNoInit</td>
<td>Next visible node in the tree without initialization.</td>
</tr>
<tr>
<td>GetNextVisibleSibling</td>
<td>Next visible sibling node with initialization.</td>
</tr>
<tr>
<td>GetNextVisibleSiblingNoInit</td>
<td>Next visible sibling node without initialization.</td>
</tr>
</tbody>
</table>

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.GetNextChecked Method
TBaseVirtualTree Class

Not documented.

Pascal

function GetNextChecked(Node: PVirtualNode; State: T)

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetNodeAt Method (Integer, Integer)

TBaseVirtualTree Class

Not documented.

Pascal

```
function GetNodeAt(X: Integer; Y: Integer): PVirtualNode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetNodeAt Method (Integer, Integer, Boolean, Integer)

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
function GetNodeAt(X: Integer; Y: Integer; Relative: Boolean; Depth: Integer): Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
**TBaseVirtualTree.GetNodeData Method**

**TBaseVirtualTree Class**

Returns the address of the user data area of the given node.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

---

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetNodeLevel Method

TBaseVirtualTree Class

Returns the indentation level of the given node.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetOptionsClass Method

TBaseVirtualTree Class

Customization helper to determine which options class the tree should use.

Pascal

```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 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: TTreeNodeOptions) begin
  FOptions.Assign(Value);
end;

//-------------------------------

function TVirtualStringTree.GetOptionsClass: TTreeOptionsClass;
begin
  Result := TStringTreeOptions;
end;

Class
  TBaseVirtualTree Class

Links
  TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetPrevious Method

TBaseVirtualTree Class

Group of node navigation functions.

Pascal

```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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetPrevious</td>
<td>Previous node in the tree with initialization.</td>
</tr>
<tr>
<td>GetPreviousInitialized</td>
<td>Previous initialized node in the tree (no initialization needed).</td>
</tr>
<tr>
<td>GetPreviousNoInit</td>
<td>Previous node in the tree without initialization.</td>
</tr>
<tr>
<td>GetPreviousSibling</td>
<td>Previous sibling node with initialization.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetPreviousVisible</td>
<td>Previous visible node in the tree with initialization.</td>
</tr>
<tr>
<td>GetPreviousVisibleNoInit</td>
<td>Previous visible node in the tree without initialization.</td>
</tr>
<tr>
<td>GetPreviousVisibleSibling</td>
<td>Previous visible sibling node with initialization.</td>
</tr>
<tr>
<td>GetPreviousVisibleSiblingNoInit</td>
<td>Previous visible sibling node without initialization.</td>
</tr>
</tbody>
</table>

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.GetSortedCutCopySet
Method
TBaseVirtualTree Class

Returns a sorted list of nodes, which are marked for cut or copy clipboard operation.

Pascal

```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

Links
   TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.GetSortedSelection Method

TBaseVirtualTree Class

Returns a sorted list of all currently selected nodes.

Pascal

```pascal
function GetSortedSelection(Resolve: Boolean): TNode;
```

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 particular 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 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
TBaseVirtualTree.GetTextInfo Method

TBaseVirtualTree Class

Helper method for node editors, hints etc.

Pascal

procedure GetTextInfo(Node: PVirtualNode; Column: TColumnIndex);

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.GetTreeFromDataObject**

**Method**

TBaseVirtualTree Class

**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.

**Pascal**

```pascal
function GetTreeFromDataObject(const DataObject: IDataObject):...
```

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.GetTreeRect Method**

*TBaseVirtualTree Class*

Returns the size of the virtual tree image.

**Pascal**

```pascal
function GetTreeRect: TRect;
```

**Description**

GetTreeRect can be used to determine the full size of the tree image as used for painting etc.

**Class**

*TBaseVirtualTree Class*

**Links**

*TBaseVirtualTree Class*

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.GetVisibleParent Method**

*TBaseVirtualTree Class*

Returns the first (nearest) parent node, which is visible.

**Pascal**

```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*

**Links**

*TBaseVirtualTree Class*
**TBaseVirtualTree.HandleHotTrack Method**

*TBaseVirtualTree Class*

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*

**Links**

*TBaseVirtualTree Class*

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.HandleIncrementalSearch Method

TBaseVirtualTree Class

Not documented.

Pascal

```
procedure HandleIncrementalSearch(CharCode: Word);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.HandleMouseDblClick Method

TBaseVirtualTree Class

Not documented.

Pascal

```
procedure HandleMouseDblClick(var Message: TWMMouse;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.HandleMouseDown
Method
TBaseVirtualTree Class

Not documented.

Pascal

procedure HandleMouseDown(var Message: TWMMouse; const Description: string);

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.HandleMouseUp Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
procedure HandleMouseUp(var Message: TWMMouse; const
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.HasAsParent Method**

**TBaseVirtualTree Class**

Determines if the given node has got another node as one of its parents.

**Pascal**

```pascal
function HasAsParent(Node: PVirtualNode; PotentialParent: PVirtualNode): Boolean;
```

**Description**

Determines whether **Node** has got **PotentialParent** as one of its parents.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.HasImage Method

TBaseVirtualTree Class

Not documented.

Pascal

function HasImage(Node: PVirtualNode; Kind: TVTImage);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.HasPopupMenu Method

Determines whether there is a pop up menu assigned to the tree.

Pascal

```
function HasPopupMenu(Node: PVirtualNode; Column: TColumnIndex): Boolean;
```

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` 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.InitChildren Method
TBaseVirtualTree Class

Not documented.

Pascal

```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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.InitNode Method
TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.InsertNode Method

TBaseVirtualTree Class

Inserts a new node and returns it to the caller.

Pascal

```
function InsertNode(Node: PVirtualNode; Mode: TVTNodeAttachMode);
```

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` and will also cause to trigger the `OnFreeNode` 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.InternalAddFromStream Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
procedure InternalAddFromStream(Stream: TStream; Version: Integer; Node: TBaseVirtualTree); external;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.InternalAddToSelection Method (PVirtualNode, Boolean)

TBaseVirtualTree Class

Not documented.

Pascal

function InternalAddToSelection(Node: PVirtualNode;)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.InternalAddToSelection Method (TNodeArray, Integer, Boolean)

TBaseVirtualTree Class

Not documented.

Pascal

function InternalAddToSelection(const NewItems: TNodeArray): Integer;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
**TBaseVirtualTree.InternalCacheNode Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.InternalClearSelection Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
### TBaseVirtualTree.InternalConnectNode Method

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
procedure InternalConnectNode(Node: PVirtualNode; Destination:
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.InternalData Method**

**TBaseVirtualTree Class | See Also**

Returns the address of the internal data for a tree class.

**Pascal**

```pascal
function InternalData(Node: PVirtualNode): Pointer;
```

**Description**

In TBaseVirtualTree 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

**Class**

- TBaseVirtualTree Class

**Links**

- TBaseVirtualTree Class, See Also

---

What do you think about this topic? Send feedback!
TBaseVirtualTree.InternalDisconnectNode Method
TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure InternalDisconnectNode(Node: PVirtualNode;
  KeepFocus: Boolean;
  Reindex: Boolean = True);
```

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.InternalRemoveFromSelection Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure InternalRemoveFromSelection(Node: PVirtual);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.InvalidateCache Method

TBaseVirtualTree Class

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 is called.

The internal node cache is used to speed up display in Virtual Treeview. It contains node references with a distance of CacheThreshold 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.InvalidateChildren Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.InvalidateColumn Method**

**TBaseVirtualTree Class**

Invalidates the client area part of a column.

**Pascal**

```pascal
procedure InvalidateColumn(Column: TColumnIndex);
```

**Description**

Invalidates the client area part of a column.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.InvalidateNode Method**

**TBaseVirtualTree Class**

Invalidates the given node.

**Pascal**

```pascal
function InvalidateNode(Node: PVirtualNode): TRect;
```

**Description**

InvalidateNode initiates repaint of the given node by calling InvalidateRect with the node's display rectangle and returns this rectangle.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.InvalidateToBottom Method

TBaseVirtualTree Class

Invalidates the client area starting with the top position of the given node.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.InvertSelection Method
TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.IsEditing Method**

**TBaseVirtualTree Class**

Tells the caller whether the tree is currently in edit mode.

**Pascal**

```pascal
function IsEditing: Boolean;
```

**Description**

Just a simple shortcut to test the tsEditing state.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.IsMouseSelecting Method

Tell the caller whether the tree is currently in draw selection mode.

Pascal

```pascal
function IsMouseSelecting: Boolean;
```

Description

IsMouseSelecting returns true if draw selection by the user is active or pending.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.TBaseVirtualTree IterateSubtree Method

Iterator method to go through all nodes of a given sub tree.

Pascal

function IterateSubtree(Node: PVirtualNode; Callback

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.
What do you think about this topic? Send feedback!
TBaseVirtualTree.Loaded Method

TBaseVirtualTree Class

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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.LoadFromFile Method

TBaseVirtualTree Class | See Also

 Loads previously streamed out tree data back in again.

Pascal

```pascal
procedure LoadFromFile(const FileName: TFileName);

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

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.MainColumnChanged Method

TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.MarkCutCopyNodes

Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.MeasureItemHeight Method
TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure MeasureItemHeight(const Canvas: TCanvas; N
```

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree(MouseMove Method)

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure MouseMove(Shift: TShiftState; X: Integer;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.MoveTo Method**

*(PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean)*

*Description*

Moves *Source* and all its child nodes to *Target*.

**Pascal**

```pascal
procedure MoveTo(Source: PVirtualNode; Target: PVirtualNode;
procedure MoveTo(Node: PVirtualNode; Tree: TBaseVirtualTree;
```

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

Links
   TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.Notification Method

Not documented.

Pascal

```
procedure Notification(AComponent: TComponent; Operation: TOperation);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.OriginalWMNCPaint Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```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**

**Links**

**TBaseVirtualTree Class**

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.Paint Method

TBaseVirtualTree Class

TControl's Paint method used here to display the tree.

Pascal

procedure Paint; override;

Description

Overridden method to paint the tree image. The actual work is however done in PaintTree.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
TBaseVirtualTree.PaintCheckImage Method

TBaseVirtualTree Class

Not documented.

Pascal

procedure PaintCheckImage(const PaintInfo: TVTPaintInfo);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.PaintImage Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure PaintImage(var PaintInfo: TVTPaintInfo; ImageInfoIndex:
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.PaintNodeButton Method
TBaseVirtualTree Class

Not documented.

Pascal

```
procedure PaintNodeButton(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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.PaintSelectionRectangle Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```
procedure PaintSelectionRectangle(Target: TCanvas; WindowOrgX: Integer);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.PaintTree Method**

**TBaseVirtualTree Class | See Also**

Main paint routine for the tree image.

**Pascal**

```pascal
procedure PaintTree(TargetCanvas: TCanvas; Window: TRect; Target: TPoint; PaintOptions: TPaintOptions);
```

**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**. **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

**Class**

  - TBaseVirtualTree Class
Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
**TBaseVirtualTree(PaintTreeLines Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
procedure PaintTreeLines(const PaintInfo: TVTPaintInfo);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.PanningWindowProc Method

TBaseVirtualTree Class

Not documented.

Pascal

```
procedure PanningWindowProc(var Message: TMessage);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.PasteFromClipboard Method

TBaseVirtualTree Class

Inserts the content of the clipboard into the tree.

Pascal

```pascal
function PasteFromClipboard: Boolean; virtual;
```

Description

PasteFromClipboard 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.PasteFromClipboard Method**

TBaseVirtualTree Class

Not documented.

**Pascal**

```pascal
procedure PasteFromClipboard(const DataObject: IDataObject);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic?* Send feedback!
TBaseVirtualTree.Print Method
TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.ProcessDrop Method

TBaseVirtualTree Class

Helper method to ease OLE drag'n drop operations.

Pascal

```
function ProcessDrop(DataObject: IDataObject; TargetNode: TBaseVirtualTree); overload;
```

Description

ProcessDrop can be used in a OnDragDrop 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 method.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.ProcessOLEData Method**

**TBaseVirtualTree Class**

Takes serialized OLE tree data and reconstructs the former structure.

**Pascal**

```pascal
function ProcessOLEData(Source: TBaseVirtualTree; DataObject: IDataObject; TargetNode: TBaseVirtualTreeTreeNodeRef; Mode: Integer; Optimized: Boolean): Boolean;
```

**Description**

ProcessOLEData 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 format is supported by the data object and the structure could be recreated, otherwise false.
What do you think about this topic? Send feedback!
TBaseVirtualTree.ReadChunk Method

TBaseVirtualTree Class

Not documented.

Pascal

function ReadChunk(Stream: TStream; Version: Integer;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.ReadNode Method
TBaseVirtualTree Class

Not documented.

Pascal

```
procedure ReadNode(Stream: TStream; Version: Integer
```

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class
TBaseVirtualTree.RedirectFontChangeEvent Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure RedirectFontChangeEvent(Canvas: TCanvas);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.ReinitChildren Method

TBaseVirtualTree Class

Forces all child nodes of Node to be reinitialized.

Pascal

```pascal
procedure ReinitChildren(Node: PVirtualNode; Recursive: Boolean);
```

Description

ReinitChildren forces all child nodes of Node to be reinitialized. If Recursive is true then also the grandchildren are reinitialized.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
**TBaseVirtualTree.ReinitNode Method**

*TBaseVirtualTree Class*

Forces a reinitialization of the given node.

**Pascal**

```pascal
procedure ReinitNode(Node: PVirtualNode; Recursive: Boolean);
```

**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*

**Links**

*TBaseVirtualTree Class*

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.RemoveFromSelection Method**

**TBaseVirtualTree Class**

Removes the given node from the current selection.

**Pascal**

```
procedure RemoveFromSelection(Node: PVirtualNode);
```

**Description**

Removes the vsSelected style from Node's states and also removes Node from the internal selection array.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.RenderOLEData Method

TBaseVirtualTree Class

Renders pending OLE data.

Pascal

```
function RenderOLEData(const FormatEtcIn: TFormatEtc):
```

Description

RenderOLEData 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

TBaseVirtualTree Class

Links

TBaseVirtualTree Class
What do you think about this topic? Send feedback!
TBaseVirtualTree.RepaintNode Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.ResetNode Method

TBaseVirtualTree Class

Resets the given node to uninitialized.

Pascal

```pascal
procedure ResetNode(Node: PVirtualNode); virtual;
```

Description

ResetNode deletes all children of `Node` and marks it as being uninitialized.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.ResetRangeAnchor Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.RestoreFontChangeEvent Method
TBaseVirtualTree Class

Not documented.

Pascal

procedure RestoreFontChangeEvent(Canvas: TCanvas);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.SaveToFile Method

TBaseVirtualTree Class | See Also

Saves the entire content of the tree into a file or stream.

Pascal

```pascal
procedure SaveToFile(const FileName: TFileName);
procedure SaveToStream(Stream: TStream; Node: PVirtualNode);
```

Description

Saves the entire content of the tree into a file or stream.

See Also

LoadFromStream, AddFromStream

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.ScrollIntoView Method

TBaseVirtualTree Class

Scrolls the tree so that the given node comes in the client area.

Pascal

```pascal
function ScrollIntoView(Node: PVirtualNode; Center: Boolean): 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 extended 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree::SelectAll Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.SelectNodes Method
TBaseVirtualTree Class

Selects a range of nodes.

Pascal

```pascal
procedure SelectNodes(StartNode: PVirtualNode; EndNode: PVirtualNode; AddOnly: Boolean);```

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.SetBiDiMode Method

TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.SetFocusedNodeAndColumn Method

TBaseVirtualTree Class

Not documented.

Pascal

```pascal
procedure SetFocusedNodeAndColumn(Node: PVirtualNode);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.SkipNode Method

TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.Sort Method

TBaseVirtualTree Class

Sorts the given node.

Pascal

```
procedure Sort(Node: PVirtualNode; Column: TColumnIndex);
```

Description

Sort sorts the child nodes of Node. The application is queried about how to sort via the OnCompareNodes 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.SortTree Method

TBaseVirtualTree Class

Sorts the entire tree view.

Pascal

```
procedure SortTree(Column: TColumnIndex; Direction:
```

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 and therefore usually not recommended.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.StartWheelPanning Method

TBaseVirtualTree Class

Not documented.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.StopWheelPanning Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.StructureChange Method
TBaseVirtualTree Class

Not documented.

Pascal

```
procedure StructureChange(Node: PVirtualNode; Reason

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TBaseVirtualTree Class

Links
TBaseVirtualTree Class
```

What do you think about this topic? Send feedback!
**TBaseVirtualTree.SuggestDropEffect Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
function SuggestDropEffect(Source: TObject; Shift: TShiftState; Pt: TPoint; AllowedEffects: Integer): Integer;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.ToggleNode Method**

**TBaseVirtualTree Class**

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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.ToggleSelection Method

TBaseVirtualTree Class

Toggles the selection state of a range of nodes.

Pascal

```pascal
procedure ToggleSelection(StartNode: PVirtualNode; EndNode: PVirtualNode);
```

Description

ToggleSelection switches 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

Links

TBaseVirtualTree Class
**TBaseVirtualTree.UnselectNodes Method**

**TBaseVirtualTree Class**

Deselects a range of nodes.

**Pascal**

```pascal
procedure UnselectNodes(StartNode: PVirtualNode; EndNode: PVirtualNode);
```

**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

**Links**

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.UpdateAction Method

TBaseVirtualTree Class

Not documented.

Pascal

function UpdateAction(Action: TBasicAction): Boolean;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.UpdateDesigner Method
TBaseVirtualTree Class

Not documented.

Pascal

```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

Links
TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.UpdateEditBounds Method**

**Class**

TBaseVirtualTree Class

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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.UpdateHeaderRect Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.UpdateScrollBars

**Method**

**TBaseVirtualTree Class**

Applies changes to the horizontal and vertical scrollbars.

**Pascal**

```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

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.UpdateWindowAndDragImage Method**

**TBaseVirtualTree Class**

Not documented.

**Pascal**

```pascal
procedure UpdateWindowAndDragImage(const Tree: TBaseVirtualTree);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.UseRightToLeftReading Method

TBaseVirtualTree Class

Helper method for right-to-left layout.

Pascal

```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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
**TBaseVirtualTree.ValidateCache Method**

TBaseVirtualTree Class | See Also

Initiates the validation of the internal node cache.

**Pascal**

```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

**Class**

TBaseVirtualTree Class

**Links**

TBaseVirtualTree Class, See Also

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.ValidateChildren Method

TBaseVirtualTree Class

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

Links

TBaseVirtualTree Class
**TBaseVirtualTree.ValidateNode Method**

**TBaseVirtualTree Class**

Validates a given node.

**Pascal**

```pascal
procedure ValidateNode(Node: PVirtualNode; Recursive
```

**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

**Links**

TBaseVirtualTree Class

---

*What do you think about this topic? Send feedback!*
**TBaseVirtualTree.ValidateNodeDataSize Method**

*TBaseVirtualTree Class*

Helper method for node data size initialization.

**Pascal**

```
procedure ValidateNodeDataSize(var Size: Integer);
```

**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` allowing so the application to compute now its data requirement.

**Class**

*TBaseVirtualTree Class*

**Links**

*TBaseVirtualTree Class*

*What do you think about this topic? Send feedback!*
TBaseVirtualTree.WndProc Method

TBaseVirtualTree Class

Redirected window procedure to do some special processing.

Pascal

```
procedure WndProc(var Message: TMessage); override;
```

Description

WndProc has been overridden 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

Links

TBaseVirtualTree Class

What do you think about this topic? Send feedback!
TBaseVirtualTree.WriteChunks Method
TBaseVirtualTree Class | See Also

Writes the core chunks for the given node to the given stream.

Pascal

```pascal
procedure WriteChunks(Stream: TStream; Node: PVirtualNode);
```

Description

WriteChunks is part of the streaming system in Virtual Treeview and writes the core chunks for `Node` into `Stream`. Descendants 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
TBaseVirtualTree Class

Links
TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBaseVirtualTree.WriteNode Method

TBaseVirtualTree Class | See Also

Writes the cover (envelop) chunk for the given node to the given stream.

Pascal

```
procedure WriteNode(Stream: TStream; Node: PVirtualNode);
```

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, WriteToStream

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? Send feedback!
TBufferedString Class

Classes | Methods | Properties | Legend

TBufferedString = class;

Group

Classes

Members

Properties

-AsString
  Not documented.

Methods

-Add
  Not documented.
-AddNewLine
  Not documented.
-Destroy
  Not documented.

Legend

- public
  Property
  read only
  Method
virtual

Class Hierarchy

File

VirtualTrees

Links

Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TBufferedString.AsString Property

Not documented.

Pascal

```
propertyAsString: string;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TBufferedString Class

Links

TBufferedString Class

What do you think about this topic? Send feedback!
TBufferedString.Add Method
TBufferedString Class

Not documented.

Pascal

```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

Links
TBufferedString Class

What do you think about this topic? Send feedback!
**TBufferedString.AddNewLine Method**

**TBufferedString Class**

Not documented.

**Pascal**

```pascal
procedure AddNewLine;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TBufferedString Class

**Links**

TBufferedString Class

---

*What do you think about this topic? Send feedback!*
**TBufferedString.Destroy Destructor**

**TBufferedString Class**

Not documented.

**Pascal**

```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

**Links**

TBufferedString Class

*What do you think about this topic? Send feedback!*
TClipboardFormatList Class

Classes | Methods | Legend

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

Members

Methods

- **Add**
  Adds the given data to the internal list.

- **Clear**
  Not documented.

- **Create**
  Not documented.

- **Destroy**
  Not documented.

- **EnumerateFormats**
  Returns a list of format records for the given class.

- **FindFormat**
  Not documented.
Legend

- public
- Method
- virtual

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Legend

What do you think about this topic? Send feedback!
**TClipboardFormatList.Add Method**

**TClipboardFormatList Class**

Adds the given data to the internal list.

**Pascal**

```pascal
procedure Add(FormatString: string; AClass: TVirtual
```

**Description**

The priority value is used to sort formats for importance. Larger priority values mean less priority.

**Class**

TClipboardFormatList Class

**Links**

TClipboardFormatList Class

*What do you think about this topic? Send feedback!*
TClipboardFormatList.Clear Method

TClipboardFormatList Class

Not documented.

Pascal

```pascal
procedure Clear;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormatList Class

Links

TClipboardFormatList Class

What do you think about this topic? Send feedback!
**TClipboardFormatList.Create Constructor**

*TClipboardFormatList Class*

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*

**Links**

*TClipboardFormatList Class*

What do you think about this topic? Send feedback!
TClipboardFormatList.Destory

Destructor

TClipboardFormatList Class

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

Links

TClipboardFormatList Class

What do you think about this topic? Send feedback!
TClipboardFormatList.EnumerateFormats Method (TVirtualTreeClass, TFormatEtcArray, TClipboardFormats)

**TClipboardFormatList Class**

Returns a list of format records for the given class.

**Pascal**

```pascal
procedure EnumerateFormats(TreeClass: TVirtualTreeClass; Description; If assigned the AllowedFormats is used to limit the enumerated formats to those described in the list.

**Class**

TClipboardFormatList Class

**Links**

TClipboardFormatList Class

What do you think about this topic? Send feedback!
TClipboardFormatList Class

TClipboardFormatList.EnumerateFormats Method

**TClipboardFormatList.EnumerateFormats** **Method** *(TVirtualTreeClass, TFormatEtcArray, TClipboardFormats)*

TClipboardFormatList Class

Returns a list of format descriptions for the given class.

Pascal

```pascal
procedure EnumerateFormats(TreeClass: TVirtualTreeClass;
                            Formats: TFormatEtcArray;
                            Clipboard: TClipboardFormats);
```

Class

TClipboardFormatList Class

Links

TClipboardFormatList Class

What do you think about this topic? Send feedback!
TClipboardFormatList.FindFormat Method (Word, string)
TClipboardFormatList Class

Not documented.

Pascal

```pascal
function FindFormat(Fmt: Word; var Description: string)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormatList Class

Links

TClipboardFormatList Class

What do you think about this topic? Send feedback!
TClipboardFormatList.FindFormat Method (string)

TClipboardFormatList Class

Not documented.

Pascal

```pascal
function FindFormat(FormatString: string): PClipboardFormatListEntry;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormatList Class

Links

TClipboardFormatList Class

What do you think about this topic? Send feedback!
TClipboardFormatList.FindFormat Method (string, Word)

TClipboardFormatList Class

Not documented.

Pascal

```pascal
function FindFormat(FormatString: string; var Fmt: Word):
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TClipboardFormatList Class

Links

TClipboardFormatList Class

What do you think about this topic? Send feedback!
**TClipboardFormats Class**

List of strings describing clipboard formats.

**Pascal**

```pascal
TClipboardFormats = class(TStringList);
```

**Description**

This class is an extended string list which allows to enter description strings for clipboard formats which are checked against registered formats and only accepted if the particular format could be found. This way there is an unambiguous and portable description of allowed clipboard formats possible.

**Group**

Classes

**Members**

**Properties**

- **Owner**
  - Not documented.

**Methods**

- **Add**
  - Adds a new format to the internal list.
Create
Constructor of the class.

Insert
Adds a new format to the internal list.

Legend

- public
- Property
- read only
- Method
- virtual

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TClipboardFormats.Owner Property

TClipboardFormats Class

Not documented.

Pascal

```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

Links

TClipboardFormats Class

What do you think about this topic? Send feedback!
TClipboardFormats.Add Method
TClipboardFormats Class

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 clipboard formats to only those which are registered with the owner tree or one of its ancestors.

Class

TClipboardFormats Class

Links

TClipboardFormats Class

What do you think about this topic? Send feedback!
TClipboardFormats.Create Constructor
TClipboardFormats Class

Constructor of the class.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree); virtual
```

Description

Create initializes the class.

Class

TClipboardFormats Class

Links

TClipboardFormats Class

What do you think about this topic? Send feedback!
TCriticalSection Class

Fields | Classes | Methods | Legend

Not documented.

Pascal

TCriticalSection = class(TObject);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes

Members

Fields

�行 FSection
Not documented.

Methods

�行 Create
Not documented.

�行 Destroy
Not documented.

�行 Enter
Not documented.

�行 Leave
Not documented.
Legend

- protected
- Data Member
- public
- Method
- virtual

Class Hierarchy

File

VirtualTrees

Links

Fields, Classes, Methods, Legend

What do you think about this topic? Send feedback!
TCriticalSection.FSection Field
TCriticalSection Class

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

Links
TCriticalSection Class

*What do you think about this topic?* Send feedback!
**TCriticalSection.Create Constructor**

**TCriticalSection Class**

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

**Links**

TCriticalSection Class

*What do you think about this topic? Send feedback!*
TCriticalSection.Destroy Destructor

TCriticalSection Class

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

Links

TCriticalSection Class

What do you think about this topic? Send feedback!
TCriticalSection.Enter Method

TCriticalSection Class

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

Links

TCriticalSection Class

What do you think about this topic? Send feedback!
TCriticalSection.Leave Method
TCriticalSection Class

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

Links
TCriticalSection Class

What do you think about this topic? Send feedback!
**TCustomStringTreeOptions Class**

**Classes** | **Methods** | **Properties** | **Legend**

Enhanced options class for string trees.

**Pascal**

```pascal
TCustomStringTreeOptions = class(TCustomVirtualTreeOptions)
```

**Description**

This class enhances the base class `TCustomVirtualTreeOptions` by options related to a string tree.

**Group**

Classes

**Members**

**Properties**

- **StringOptions**
  The new options introduced by the class.

**TCustomVirtualTreeOptions Class**

- **AnimationOptions**
  Options related to animations.
- **AutoOptions**
  Options related to automatic actions.
- **MiscOptions**
  Options not related to any other category.
- **Owner**
  Owner tree to which the property class belongs.
PaintOptions
Options related to painting.

SelectionOptions
Options related to the way nodes can be selected.

Methods

AssignTo
Used to copy the options class.

Create
The constructor of the class.

TCustomVirtualTreeOptions Class

AssignTo
Used to copy this option class to another option collection.

Create
Constructor of the class.

Legend

protected

Property

public

read only

Method

virtual

Class Hierarchy

File

VirtualTrees
What do you think about this topic? Send feedback!
**TCustomStringTreeOptions.StringOptions Property**

**TCustomStringTreeOptions Class**

The new options introduced by the class.

**Pascal**

```pascal
property StringOptions: TVTStringOptions;
```

**Description**

StringOptions provides access to the newly introduced options by which the base class is extended.

**Class**

**TCustomStringTreeOptions Class**

**Links**

**TCustomStringTreeOptions Class**

---

*What do you think about this topic? Send feedback!*
TCustomStringTreeOptions.AssignTo Method

Used to copy the options class.

Pascal

```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

Links

TCustomStringTreeOptions Class

What do you think about this topic? Send feedback!
**TCustomStringTreeOptions.Create**  
*Constructor*

The constructor of the class.

**Pascal**

```
constructor Create(AOwner: TBaseVirtualTree); override
```

**Description**

The constructor initializes the class.

**Class**

TCustomStringTreeOptions Class

**Links**

TCustomStringTreeOptions Class
TCustomVirtualDrawTree Class

Simple owner draw descendant of the base tree.

Pascal

```pascal
TCustomVirtualDrawTree = class(TBaseVirtualTree);
```

Description

TCustomVirtualDrawTree is a simple TBaseVirtualTree descendant, which publishes the paint method through an event. This allows an application for self drawn tree views without overriding the base class.

Group

Classes

Members

Events

- **OnDrawHint**
  Triggered when a node hint or tooltip must be drawn.

- **OnDrawNode**
  Triggered when a node must be drawn.

- **OnGetHintSize**
  Triggered when a node hint or tooltip is about to show.

- **OnGetNodeWidth**
  Triggered when a node is about to be drawn.

TBaseVirtualTree Class
OnAdvancedHeaderDraw
  Header paint support event.
OnAfterCellPaint
  Paint support event.
OnAfterItemErase
  Paint support event.
OnAfterItemPaint
  Paint support event.
OnAfterPaint
  Paint support event.
OnBeforeCellPaint
  Paint support event.
OnBeforeItemErase
  Paint support event.
OnBeforeItemPaint
  Paint support event.
OnBeforePaint
  Paint support event.
OnChange
  Navigation support event.
OnChecked
  Check support event.
OnChecking
  Check support event.
OnCollapsed
  Miscellaneous event.
OnCollapsing
  Miscellaneous event.
OnColumnClick
  Header and column support event.
OnColumnDblClick
  Header and column support event.
OnColumnResize
  Header and column support routine.
OnCompareNodes
Sort and search support event.

- **OnCreateDataObject**
  Drag'n drop support event.

- **OnCreateDragManager**
  Drag'n drop support event.

- **OnCreateEditor**
  Editing support event.

- **OnDragAllowed**
  Drag'n drop support event.

- **OnDragDrop**
  Drag'n drop support event.

- **OnDragOver**
  Drag'n drop support event.

- **OnEditCancelled**
  Editing support event.

- **OnEdited**
  Editing support event.

- **OnEditing**
  Editing support event.

- **OnExpanded**
  Miscellaneous event.

- **OnExpanding**
  Miscellaneous event.

- **OnFocusChanged**
  Navigation support event.

- **OnFocusChanging**
  Navigation support event.

- **OnFreeNode**
  Data management node.

- **OnGetCellIsEmpty**
  Triggered when the tree control needs to know whether a given column is empty.

- **OnGetCursor**
  Miscellaneous event.

- **OnGetHeaderCursor**
Header and column support event.

- **OnGetHelpContext**
  Miscellaneous event.

- **OnGetImageIndex**
  Display management event.

- **OnGetImageIndexEx**
  Not documented.

- **OnGetLineStyle**
  Display management event.

- **OnGetNodeDataSize**
  Data management event.

- **OnGetPopupMenu**
  Miscellaneous event.

- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.

- **OnHeaderClick**
  Header & column support event.

- **OnHeaderDb1Click**
  Header & column support event.

- **OnHeaderDragged**
  Header & column support event.

- **OnHeaderDraggedOut**
  Header & column support event.

- **OnHeaderDragging**
  Header & column support event.

- **OnHeaderDraw**
  Header & column support event.

- **OnHeaderDrawQueryElements**
  Header & column support event.

- **OnHeaderMouseDown**
  Header & column support event.

- **OnHeaderMouseMove**
  Header & column support event.

- **OnHeaderMouseUp**
  Header & column support event.
OnHotChange
Navigation support event.

OnIncrementalSearch
Miscellaneous event.

OnInitChildren
Node management event.

OnInitNode
Node management event.

OnKeyAction
Miscellaneous event.

OnLoadNode
Streaming support event.

OnMeasureItem
Miscellaneous event.

OnNodeCopied
Miscellaneous event.

OnNodeCopying
Miscellaneous event.

OnNodeMoved
Miscellaneous event.

OnNodeMoving
Miscellaneous event.

OnPaintBackground
Paint support event.

OnRenderOLEData
Drag’n drop and clipboard support event.

OnResetNode
Node management event.

OnSaveNode
Streaming support event.

OnScroll
Miscellaneous event.

OnShowScrollbar
Not documented.

OnStateChange
Miscellaneous event.

**OnStructureChange**
Miscellaneous event.

**OnUpdating**
Miscellaneous event.

### Methods

- **DoDrawHint**
  Overridable method which triggers **OnDrawHint**.
- **DoGetHintSize**
  Overridable method which triggers **OnGetHintSize**.
- **DoGetNodeWidth**
  Overridable method which triggers **OnGetNodeWidth**.
- **DoPaintNode**
  Overridable method which triggers **OnPaintNode**.

### TBaseVirtualTree Class

- **AbsoluteIndex**
  Reads the overall index of a node.
- **AddChild**
  Creates and adds a new child node to given node.
- **AddFromStream**
  Adds the content from the given stream to the given node.
- **AddToSelection**
  Adds one or more nodes to the current selection.
- **AdjustPaintCellRect**
  Used in descendants to modify the clip rectangle of the current column while painting a certain node.
- **AdjustPanningCursor**
  Loads the proper cursor which indicates into which direction scrolling is done.
- **AdviseChangeEvent**
  Used to register a delayed change event.
- **AllocateInternalDataArea**
Registration method to allocate tree internal data per node.

Animate
Support method for animated actions in the tree view.

Assign
Used to copy properties from another Virtual Treeview.

BeginDrag
Starts an OLE drag'n drop operation.

BeginSynch
Enters the tree into a special synchronized mode.

BeginUpdate
Locks the tree view to perform several update operations.

CalculateSelectionRect
Support method for draw selection.

CanAutoScroll
Determines whether the tree can currently auto scroll its window.

CancelCutOrCopy
Cancels any pending cut or copy clipboard operation.

CancelEditNode
Cancel the current edit operation, if there is any.

CanEdit
Determines whether a node can be edited or not.

CanFocus
Support method to determine whether the tree window can receive the input focus.

CanShowDragImage
Determines whether a drag image should be shown.

Change
Central method called when a node's selection state changes.

ChangeScale
Helper method called by the VCL when control resizing is due.

CheckParentCheckState
Helper method for recursive check state changes.

Clear
Clears the tree and removes all nodes.

ClearChecked
Not documented.

**ClearSelection**
Removes all nodes from the current selection.

**ClearTempCache**
Helper method to clear the internal temporary node cache.

**ColumnIsEmpty**
Used to determine if a cell is considered as being empty.

**CopyTo**
Copies **Source** and all its child nodes to **Target**.

**CopyToClipBoard**
Copies all currently selected nodes to the clipboard.

**CountLevelDifference**
Determines the level difference of two nodes.

**CountVisibleChildren**
Determines the number of visible child nodes of the given node.

**Create**
Constructor of the control

**CreateParams**
Prepares the creation of the controls window handle.

**CreateWnd**
Initializes data, which depends on the window handle.

**CutToClipBoard**
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

**DefineProperties**
Helper method to customize loading and saving persistent tree data.

**DeleteChildren**
Removes all child nodes from the given node.

**DeleteNode**
Removes the given node from the tree.

**DeleteSelectedNodes**
Removes all currently selected nodes form the tree.

**Destroy**
Destructor of the control.
Determines whether all children of a given node are hidden.

**DetermineHiddenChildrenFlagAllNodes**
Determines whether all children of all nodes are hidden.

**DetermineHitPositionLTR**
Determines the hit position within a node with left-to-right and right-to-left orientation.

**DetermineHitPositionRTL**
Determines the hit position within a node with left-to-right and right-to-left orientation.

**DetermineNextCheckState**
Not documented.

**DetermineScrollDirections**
Not documented.

**DoAdvancedHeaderDraw**
Not documented.

**DoAfterCellPaint**
Not documented.

**DoAfterItemErase**
Not documented.

**DoAfterItemPaint**
Not documented.

**DoAfterPaint**
Not documented.

**DoAutoScroll**
Enables or disables the auto scroll timer.

**DoBeforeCellPaint**
Not documented.

**DoBeforeDrag**
Not documented.

**DoBeforeItemErase**
Not documented.

**DoBeforeItemPaint**
Not documented.

**DoBeforePaint**
Not documented.
DoCancelEdit
Called when the tree should stop editing without accepting changed values.

DoCanEdit
Not documented.

DoChange
Not documented.

DoCheckClick
Not documented.

DoChecked
Not documented.

DoChecking
Not documented.

DoCollapsed
Not documented.

DoCollapsing
Not documented.

DoColumnClick
Not documented.

DoColumnDblClick
Not documented.

DoColumnResize
Not documented.

DoCompare
Not documented.

DoCreateDataObject
Not documented.

DoCreateDragManager
Not documented.

DoCreateEditor
Not documented.

DoDragDrop
Not documented.

DoDragExpand
Not documented.
DoDragging
Internal method which handles drag' drop.

DoDragOver
Not documented.

DoEdit
Initiates editing of the currently set focused column and edit node.

DoEndDrag
Not documented.

DoEndEdit
Stops the current edit operation and takes over the new content.

DoExpanded
Not documented.

DoExpanding
Not documented.

DoFocusChange
Not documented.

DoFocusChanging
Not documented.

DoFocusNode
Internal method to set the focused node.

DoFreeNode
Not documented.

DoGetAnimationType
Determines the type of animation to be used.

DoGetCursor
Not documented.

DoGetHeaderCursor
Not documented.

DoGetImageIndex
Not documented.

DoGetLineStyle
Not documented.

DoGetNodeHint
Not documented.

DoGetNodeTooltip
Not documented.

- **DoGetNodeWidth**
  Overridable method which always returns 0.

- **DoGetPopupMenu**
  Overridable method which triggers the OnGetPopup event.

- **DoGetUserClipboardFormats**
  Not documented.

- **DoHeaderClick**
  Not documented.

- **DoHeaderDb1Click**
  Not documented.

- **DoHeaderDragged**
  Not documented.

- **DoHeaderDraggedOut**
  Not documented.

- **DoHeaderDragging**
  Not documented.

- **DoHeaderDraw**
  Not documented.

- **DoHeaderDrawQueryElements**
  Not documented.

- **DoHeaderMouseDown**
  Not documented.

- **DoHeaderMouseMove**
  Not documented.

- **DoHeaderMouseUp**
  Not documented.

- **DoHotChange**
  Not documented.

- **DoIncrementalSearch**
  Not documented.

- **DoInitChildren**
  Not documented.

- **DoInitNode**
  Not documented.
DoKeyAction
Not documented.

DoLoadUserData
Not documented.

DoMeasureItem
Not documented.

DoNodeCopied
Not documented.

DoNodeCopying
Not documented.

DoNodeMoved
Not documented.

DoNodeMoving
Not documented.

DoPaintBackground
Not documented.

DoPaintDropMark
Overridable method which draws the small line on top of a node's image depending on the current drop state.

DoPaintNode
Overridable method which does nothing.

DoPopupMenu
Overridable method which shows the popup menu for the given node.

DoRenderOLEData
Not documented.

DoReset
Not documented.

DoSaveUserData
Not documented.

DoScroll
Overridable method which triggers the OnScroll event.

DoSetOffsetXY
Internal core routine to set the tree's scroll position.

DoShowScrollbar
Not documented.
DoStartDrag
Not documented.

DoStateChange
Not documented.

DoStructureChange
Not documented.

DoTimerScroll
Callback method which is triggered whenever the scroll timer fires.

DoUpdating
Not documented.

DoValidateCache
Not documented.

DragCanceled
Called by the VCL when a drag'n drop operation was canceled by the user.

DragDrop
Helper method, which is used when a drag operation is finished.

DragEnter
Not documented.

DragFinished
Called when a drag operation is finished (accepted or cancelled).

Dragging
Returns true if a drag'n drop operation is in progress.

DragLeave
Not documented.

DragOver
Not documented.

DrawDottedHLine
Not documented.

DrawDottedVLine
Not documented.

EditNode
Starts editing the given node if allowed to.

EndEditNode
Stops node editing if it was started before.
EndSynch
Counterpart to BeginSynch.
EndUpdate
Resets the update lock set by BeginUpdate.
ExecuteAction
Not documented.
FindNodeInSelection
Helper method to find the given node in the current selection.
FinishChunkHeader
Not documented.
FinishCutOrCopy
Stops any pending cut or copy clipboard operation.
FlushClipboard
Renders all pending clipboard data.
FontChanged
Not documented.
FullCollapse
Collapses all nodes in the tree.
FullExpand
Expands all nodes in the tree.
GetBorderDimensions
Not documented.
GetCheckImage
Not documented.
GetCheckImageListFor
Not documented.
GetColumnClass
Returns the class to be used to manage columns in the tree.
GetControlsAlignment
Not documented.
GetDisplayRect
Returns the visible region used by the given node in client coordinates.
GetFirst
Group of node navigation functions.
GetFirstChild
Group of node navigation functions.

GetFirstCutCopy
Group of node navigation functions.

GetFirstInitialized
Group of node navigation functions.

GetFirstNoInit
Group of node navigation functions.

GetFirstSelected
Group of node navigation functions.

GetFirstVisible
Group of node navigation functions.

GetFirstVisibleChild
Group of node navigation functions.

GetFirstVisibleChildNoInit
Group of node navigation functions.

GetFirstVisibleNoInit
Group of node navigation functions.

GetHeaderClass
Returns the header class to be used by the tree.

GetHintWindowClass
Not documented.

GetHitTestInfoAt
Returns information about the node at the given position.

GetImageIndex
Not documented.

GetLast
Group of node navigation functions.

GetLastChild
Group of node navigation functions.

GetLastChildNoInit
Group of node navigation functions.

GetLastInitialized
Group of node navigation functions.

- **GetLastNoInit**
  Group of node navigation functions.

- **GetLastVisible**
  Group of node navigation functions.

- **GetLastVisibleChild**
  Group of node navigation functions.

- **GetLastVisibleChildNoInit**
  Group of node navigation functions.

- **GetLastVisibleNoInit**
  Group of node navigation functions.

- **GetMaxColumnWidth**
  Returns the width of the largest node in the given column.

- **GetMaxRightExtend**
  Determines the maximum with of the currently visible part of the tree.

- **GetNativeClipboardFormats**
  Used to let descendants and the application add their own supported clipboard formats.

- **GetNext**
  Group of node navigation functions.

- **GetNextChecked**
  Not documented.

- **GetNextCutCopy**
  Group of node navigation functions.

- **GetNextInitialized**
  Group of node navigation functions.

- **GetNextNoInit**
  Group of node navigation functions.

- **GetNextSelected**
  Group of node navigation functions.

- **GetNextSibling**
  Group of node navigation functions.

- **GetNextVisible**
  Group of node navigation functions.

- **GetNextVisibleNoInit**
Group of node navigation functions.

- **GetNextVisibleSibling**
  Group of node navigation functions.

- **GetNextVisibleSiblingNoInit**
  Group of node navigation functions.

- **GetNodeAt**
  Not documented.

- **GetNodeData**
  Returns the address of the user data area of the given node.

- **GetNodeLevel**
  Returns the indentation level of the given node.

- **GetOptionsClass**
  Customization helper to determine which options class the tree should use.

- **GetPrevious**
  Group of node navigation functions.

- **GetPreviousInitialized**
  Group of node navigation functions.

- **GetPreviousNoInit**
  Group of node navigation functions.

- **GetPreviousSibling**
  Group of node navigation functions.

- **GetPreviousVisible**
  Group of node navigation functions.

- **GetPreviousVisibleNoInit**
  Group of node navigation functions.

- **GetPreviousVisibleSibling**
  Group of node navigation functions.

- **GetPreviousVisibleSiblingNoInit**
  Group of node navigation functions.

- **GetSortedCutCopySet**
  Returns a sorted list of nodes, which are marked for cut or copy clipboard operation.

- **GetSortedSelection**
  Returns a sorted list of all currently selected nodes.
**GetTextInfo**
Helper method for node editors, hints etc.

**GetTreeFromDataObject**
OLE drag'n drop and clipboard support method.

**GetTreeRect**
Returns the size of the virtual tree image.

**GetVisibleParent**
Returns the first (nearest) parent node, which is visible.

**HandleHotTrack**
Not documented.

**HandleIncrementalSearch**
Not documented.

**HandleMouseDbClick**
Not documented.

**HandleMouseDown**
Not documented.

**HandleMouseUp**
Not documented.

**HasAsParent**
Determines if the given node has got another node as one of its parents.

**HasImage**
Not documented.

**HasPopupMenu**
Determines whether there is a pop up menu assigned to the tree.

**InitChildren**
Not documented.

**InitNode**
Not documented.

**InsertNode**
Inserts a new node and returns it to the caller.

**InternalAddFromStream**
Not documented.

**InternalAddToSelection**
Not documented.
InternalCacheNode
Not documented.

InternalClearSelection
Not documented.

InternalConnectNode
Not documented.

InternalData
Returns the address of the internal data for a tree class.

InternalDisconnectNode
Not documented.

InternalRemoveFromSelection
Not documented.

InvalidateCache
Empties the internal node cache and marks it as invalid.

InvalidateChildren
Invalidates all children of the given node.

InvalidateColumn
Invalidates the client area part of a column.

InvalidateNode
Invalidates the given node.

InvalidateToBottom
Invalidates the client area starting with the top position of the given node.

InvertSelection
Inverts the current selection.

IsEditing
Tells the caller whether the tree is currently in edit mode.

IsMouseSelecting
Tell the caller whether the tree is currently in draw selection mode.

IterateSubtree
Iterator method to go through all nodes of a given sub tree.

Loaded
Not documented.

LoadFromFile
Loads previously streamed out tree data back in again.
LoadFromStream
Loads previously streamed out tree data back in again.
MainColumnChanged
Not documented.
MarkCutCopyNodes
Not documented.
MeasureItemHeight
Not documented.
MouseMove
Not documented.
MoveTo
Moves **Source** and all its child nodes to **Target**.
Notification
Not documented.
OriginalWMNCPaint
Not documented.
Paint
TControl's Paint method used here to display the tree.
PaintCheckImage
Not documented.
PaintImage
Not documented.
PaintNodeButton
Not documented.
PaintSelectionRectangle
Not documented.
PaintTree
Main paint routine for the tree image.
PaintTreeLines
Not documented.
PanningWindowProc
Not documented.
PasteFromClipboard
Inserts the content of the clipboard into the tree.
PrepareDragImage
Not documented.

- **Print**
  Not documented.

- **ProcessDrop**
  Helper method to ease OLE drag'n drop operations.

- **ProcessOLEData**
  Takes serialized OLE tree data and reconstructs the former structure.

- **ReadChunk**
  Not documented.

- **ReadNode**
  Not documented.

- **RedirectFontChangeEvent**
  Not documented.

- **ReinitChildren**
  Forces all child nodes of Node to be reinitialized.

- **ReinitNode**
  Forces a reinitialization of the given node.

- **RemoveFromSelection**
  Removes the given node from the current selection.

- **RenderOLEData**
  Renders pending OLE data.

- **RepaintNode**
  Causes the treeview to repaint the given node.

- **ResetNode**
  Resets the given node to uninitialized.

- **ResetRangeAnchor**
  Not documented.

- **RestoreFontChangeEvent**
  Not documented.

- **SaveToFile**
  Saves the entire content of the tree into a file or stream.

- **SaveToStream**
  Saves the entire content of the tree into a file or stream.

- **ScrollIntoView**
  Scrolls the tree so that the given node comes in the client area.
SelectAll
Selects all nodes in the tree.

SelectNodes
Selects a range of nodes.

SetBiDiMode
Not documented.

SetFocusedNodeAndColumn
Not documented.

SkipNode
Not documented.

Sort
Sorts the given node.

SortTree
Sorts the entire tree view.

StartWheelPanning
Not documented.

StopWheelPanning
Not documented.

StructureChange
Not documented.

SuggestDropEffect
Not documented.

ToggleNode
Changes a node's expand state to the opposite state.

ToggleSelection
Toggles the selection state of a range of nodes.

UnselectNodes
Deselects a range of nodes.

UpdateAction
Not documented.

UpdateDesigner
Not documented.

UpdateEditBounds
Not documented.

UpdateHeaderRect
Not documented.

- **UpdateHorizontalScrollBar**
  Applies changes to the horizontal and vertical scrollbars.

- **UpdateScrollBars**
  Applies changes to the horizontal and vertical scrollbars.

- **UpdateVerticalScrollBar**
  Applies changes to the horizontal and vertical scrollbars.

- **UpdateWindowAndDragImage**
  Not documented.

- **UseRightToLeftReading**
  Helper method for right-to-left layout.

- **ValidateCache**
  Initiates the validation of the internal node cache.

- **ValidateChildren**
  Validates all children of a given node.

- **ValidateNode**
  Validates a given node.

- **ValidateNodeDataSize**
  Helper method for node data size initialization.

- **WndProc**
  Redirected window procedure to do some special processing.

- **WriteChunks**
  Writes the core chunks for the given node to the given stream.

- **WriteNode**
  Writes the cover (envelop) chunk for the given node to the given stream.

### Properties

**TBaseVirtualTree Class**

- **Alignment**
  Determines the horizontal alignment of text if no columns are defined.

- **AnimationDuration**
  Determines the maximum duration the tree can use to play an animation.
- **AutoExpandDelay**
  Time delay after which a node gets expanded if it is the current drop target.

- **AutoScrollDelay**
  Time which determines when auto scrolling should start.

- **AutoScrollInterval**
  Time interval between scroll events when doing auto scroll.

- **Background**
  Holds a background image for the tree.

- **BackgroundOffsetX**
  Horizontal offset of the background image.

- **BackgroundOffsetY**
  Vertical offset of the background image.

- **BorderStyle**
  Same as TForm.BorderStyle.

- **ButtonFillMode**
  Determines how to fill the background of the node buttons.

- **ButtonStyle**
  Determines the look of node buttons.

- **ChangeDelay**
  Time which determines when the **OnChange** event should be triggered after the actual change event.

- **CheckImageKind**
  Determines which images should be used for checkboxes and radio buttons.

- **CheckImages**
  Not documented.

- **CheckState**
  Read or set the check state of a node.

- **CheckType**
  Read or set the check type of a node.

- **ChildCount**
  Read or set the number of child nodes of a node.

- **ChildrenInitialized**
  Read whether a node's child count has been initialized already.
ClipboardFormats
Special class to keep a list of clipboard format descriptions.

Colors
A collection of colors used in the tree.

CustomCheckImages
Assign your own image list to get the check images you like most.

DefaultNodeHeight
Read or set the height new nodes get as initial value.

DefaultPasteMode
Read or set the value, which determines where to add pasted nodes to.

DragHeight
Read or set the vertical limit of the internal drag image.

DragImage
Holds the instance of the internal drag image.

DragImageKind
Read or set what should be shown in the drag image.

DragManager
Holds the reference to the internal drag manager.

DragOperations
Read or set which drag operations may be allowed in the tree.

DragSelection
Keeps a temporary list of nodes during drag'n drop.

DragType
Read or set which subsystem should be used for dragging.

DragWidth
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode
Read or set how multiselection with the mouse is to be visualized.

DropTargetNode
Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

EditColumn
Not documented.

EditDelay
Read or set the maximum time between two single clicks on the same node, which should start node editing.

- **EditLink**
  Keeps a reference to the internal edit link during a node edit operation.

- **Expanded**
  Read or set the expanded state of a particular node.

- **FocusedColumn**
  Read or set the currently focused column.

- **FocusedNode**
  Read or set the currently focused node.

- **Font**
  Same as TWinControl.Font.

- **FullyVisible**
  Read or set whether a node is fully visible or not.

- **HasChildren**
  Read or set whether a node has got children.

- **Header**
  Provides access to the header instance.

- **HeaderRect**
  Returns the non-client-area rectangle used for the header.

- **HintAnimation**
  Read or set the current hint animation type.

- **HintMode**
  Read or set what type of hint you want for the tree view.

- **HotCursor**
  Read or set which cursor should be used for hot nodes.

- **HotNode**
  Read, which node is currently the hot node.

- **Images**
  Read or set the tree's normal image list.

- **IncrementalSearch**
  Read or set the current incremental search mode.

- **IncrementalSearchDirection**
  Read or set the direction to be used for incremental search.
**IncrementalSearchStart**
Read or set where to start incremental search.

**IncrementalSearchTimeout**
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

**Indent**
Read or set the indentation amount for node levels.

**IsDisabled**
Read or set the enabled state of the given node.

**IsVisible**
Read or set the visibility state of the given node.

**LastClickPos**
Used for retained drag start and wheel mouse scrolling.

**LastDropMode**
Read how the last drop operation finished.

**LineMode**
Read or set the mode of the tree lines.

**LineStyle**
Read or set the mode of the tree lines.

**Margin**
Read or set the tree's node margin.

**MultiLine**
Read or toggle the multiline feature for a given node.

**NodeAlignment**
Read or set the node alignment value.

**NodeDataSize**
Read or set the extra data size for each node.

**NodeHeight**
Read or set a node's height.

**NodeParent**
Read or set a node's parent node.

**OffsetX**
Read or set the tree's current horizontal and vertical scroll offsets.

**OffsetXY**
Read or set the tree's current horizontal and vertical scroll offsets.
OffsetY
Read or set the tree's current horizontal and vertical scroll offsets.

RootNode
Reference to the internal root node which is the anchor of the entire tree node hierarchy.

RootNodeCount
Read or set the number of nodes on the top level.

ScrollBarOptions
Reference to the scroll bar options class.

SearchBuffer
Current input string for incremental search.

Selected
Property to modify or determine the selection state of a node.

SelectedCount
Contains the number of selected nodes.

SelectionBlendFactor
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

SelectionCurveRadius
Read or set the current corner radius for node selection rectangles.

StateImages
Reference to the images list which is used for the state images.

TextMargin
Read or set the distance of the node caption to its borders.

TopNode
The top node is the node which is currently at the top border of the client area.

TotalCount
Returns the number of nodes in the tree.

TotalInternalDataSize
Keeps the currently accumulated data size for one node.

TreeOptions
Reference to the tree's options.

TreeStates
Property which keeps a set of flags which indicate current operation
and states of the tree.

- **UpdateCount**
  Not documented.

- **VerticalAlignment**
  Used to set a node's vertical button alignment with regard to the entire node rectangle.

- **VisibleCount**
  Number of currently visible nodes.

- **VisiblePath**
  Property to set or determine a node parent’s expand states.

- **WantTabs**
  Read or set whether the tree wants to process tabs on its own.

**Legend**

- protected
- Event
- Method
- virtual
- public
- Property
- read only

**Class Hierarchy**

```
TCustomControl --> TBaseVirtualTree --> TCustomVirtualDrawTree
```

**File**

VirtualTrees

**Links**

Events, Classes, Methods, Properties, Legend
What do you think about this topic? Send feedback!
TCustomVirtualDrawTree.OnDrawHint Event

TCustomVirtualDrawTree Class

Triggered when a node hint or tooltip must be drawn.

Pascal

```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

TCustomVirtualDrawTree Class

Links

TCustomVirtualDrawTree Class
TCustomVirtualDrawTree.OnDrawNode Event

Triggered when a node must be drawn.

Pascal

```pascal
property OnDrawNode: TVTDrawNodeEvent;
```

Description

Use an event handler for OnDrawNode to draw the actual content for the given node.

Class

TCustomVirtualDrawTree Class

Links

TCustomVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TCustomVirtualDrawTree.OnGetHintSize Event**

*TCustomVirtualDrawTree Class*

Triggered when a node hint or tooltip is about to show.

**Pascal**

```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**

*TCustomVirtualDrawTree Class*

**Links**

*TCustomVirtualDrawTree Class*

What do you think about this topic? Send feedback!
**TCustomVirtualDrawTree.OnGetNodeWidth Event**

*TCustomVirtualDrawTree Class*

Triggered when a node is about to be drawn.

**Pascal**

```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*

**Links**

*TCustomVirtualDrawTree Class*

*What do you think about this topic? Send feedback!*
TCustomVirtualDrawTree.DoDrawHint

Method

TCustomVirtualDrawTree Class

Overridable method which triggers OnDrawHint.

Pascal

```
procedure DoDrawHint(Canvas: TCanvas; Node: PVirtual);
```

Description

You can override DoDrawHint to customize the behavior for this request.

Class

TCustomVirtualDrawTree Class

Links

TCustomVirtualDrawTree Class

What do you think about this topic? Send feedback!
TCustomVirtualDrawTree.DoGetHintSize Method

**Class**
TCustomVirtualDrawTree Class

Overridable method which triggers **OnGetHintSize**.

**Pascal**

```
procedure DoGetHintSize(Node: PVirtualNode; Column:...
```

**Description**

You can override **OnGetHintSize** to customize the behavior for this request.

**Class**

TCustomVirtualDrawTree Class

**Links**

TCustomVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TCustomVirtualDrawTree.DoGetNodeWidth Method**

**TCustomVirtualDrawTree Class**

Overridable method which triggers `OnGetNodeWidth`.

**Pascal**

```pascal
function DoGetNodeWidth(Node: PVirtualNode; Column: TColumnIndex): integer;
```

**Description**

You can override `OnGetNodeWidth` to customize the behavior for this request.

**Class**

**TCustomVirtualDrawTree Class**

**Links**

**TCustomVirtualDrawTree Class**

---

*What do you think about this topic? Send feedback!*
TCustomVirtualDrawTree.DoPaintNode Method

Overridable method which triggers OnPaintNode.

Pascal

```
procedure DoPaintNode(var PaintInfo: TVTPaintInfo);
```

Description

You can override OnPaintNode to customize the behavior for this request.

Class

TCustomVirtualDrawTree Class

Links

TCustomVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TCustomVirtualStringTree Class**

**Events | Classes | Methods | Properties | Legend**

Descendant of **TBaseVirtualTree**, which is able to manage node captions on its own

**Pascal**

```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.

**Group**

- **Classes**

**Members**

**Properties**

- **DefaultText**
  - Not documented.
- **EllipsisWidth**
  - Not documented.
- **Text**
  - Not documented.
- **TreeOptions**
  - Reference to the tree's options.

**TBaseVirtualTree Class**
Alignment
Determines the horizontal alignment of text if no columns are defined.

AnimationDuration
Determines the maximum duration the tree can use to play an animation.

AutoExpandDelay
Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay
Time which determines when auto scrolling should start.

AutoScrollInterval
Time interval between scroll events when doing auto scroll.

Background
Holds a background image for the tree.

BackgroundOffsetX
Horizontal offset of the background image.

BackgroundOffsetY
Vertical offset of the background image.

BorderStyle
Same as TForm.BorderStyle.

ButtonFillMode
Determines how to fill the background of the node buttons.

ButtonStyle
Determines the look of node buttons.

ChangeDelay
Time which determines when the OnChange event should be triggered after the actual change event.

CheckImageKind
Determines which images should be used for checkboxes and radio buttons.

CheckImages
Not documented.

CheckState
Read or set the check state of a node.

CheckType
Read or set the check type of a node.

**ChildCount**
Read or set the number of child nodes of a node.

**ChildrenInitialized**
Read whether a node's child count has been initialized already.

**ClipboardFormats**
Special class to keep a list of clipboard format descriptions.

**Colors**
A collection of colors used in the tree.

**CustomCheckImages**
Assign your own image list to get the check images you like most.

**DefaultNodeHeight**
Read or set the height new nodes get as initial value.

**DefaultPasteMode**
Read or set the value, which determines where to add pasted nodes to.

**DragHeight**
Read or set the vertical limit of the internal drag image.

**DragImage**
Holds the instance of the internal drag image.

**DragImageKind**
Read or set what should be shown in the drag image.

**DragManager**
Holds the reference to the internal drag manager.

**DragOperations**
Read or set which drag operations may be allowed in the tree.

**DragSelection**
Keeps a temporary list of nodes during drag'n drop.

**DragType**
Read or set which subsystem should be used for dragging.

**DragWidth**
Read or set the horizontal limit of the internal drag image.

**DrawSelectionMode**
Read or set how multiselection with the mouse is to be visualized.

**DropTargetNode**
Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

- **EditColumn**
  Not documented.

- **EditDelay**
  Read or set the maximum time between two single clicks on the same node, which should start node editing.

- **EditLink**
  Keeps a reference to the internal edit link during a node edit operation.

- **Expanded**
  Read or set the expanded state of a particular node.

- **FocusedColumn**
  Read or set the currently focused column.

- **FocusedNode**
  Read or set the currently focused node.

- **Font**
  Same as TWinControl.Font.

- **FullyVisible**
  Read or set whether a node is fully visible or not.

- **HasChildren**
  Read or set whether a node has got children.

- **Header**
  Provides access to the header instance.

- **HeaderRect**
  Returns the non-client-area rectangle used for the header.

- **HintAnimation**
  Read or set the current hint animation type.

- **HintMode**
  Read or set what type of hint you want for the tree view.

- **HotCursor**
  Read or set which cursor should be used for hot nodes.

- **HotNode**
  Read, which node is currently the hot node.

- **Images**
Read or set the tree's normal image list.

- **IncrementalSearch**
  Read or set the current incremental search mode.

- **IncrementalSearchDirection**
  Read or set the direction to be used for incremental search.

- **IncrementalSearchStart**
  Read or set where to start incremental search.

- **IncrementalSearchTimeout**
  Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

- **Indent**
  Read or set the indentation amount for node levels.

- **IsDisabled**
  Read or set the enabled state of the given node.

- **IsVisible**
  Read or set the visibility state of the given node.

- **LastClickPos**
  Used for retained drag start and wheel mouse scrolling.

- **LastDropMode**
  Read how the last drop operation finished.

- **LineMode**
  Read or set the mode of the tree lines.

- **LineStyle**
  Read or set the mode of the tree lines.

- **Margin**
  Read or set the tree's node margin.

- **MultiLine**
  Read or toggle the multiline feature for a given node.

- **NodeAlignment**
  Read or set the node alignment value.

- **NodeDataSize**
  Read or set the extra data size for each node.

- **NodeHeight**
  Read or set a node's height.

- **NodeParent**
Read or set a node's parent node.

- **OffsetX**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetXY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **RootNode**
  Reference to the internal root node which is the anchor of the entire tree node hierarchy.

- **RootNodeCount**
  Read or set the number of nodes on the top level.

- **ScrollBarOptions**
  Reference to the scroll bar options class.

- **SearchBuffer**
  Current input string for incremental search.

- **Selected**
  Property to modify or determine the selection state of a node.

- **SelectedCount**
  Contains the number of selected nodes.

- **SelectionBlendFactor**
  Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

- **SelectionCurveRadius**
  Read or set the current corner radius for node selection rectangles.

- **StateImages**
  Reference to the images list which is used for the state images.

- **TextMargin**
  Read or set the distance of the node caption to its borders.

- **TopNode**
  The top node is the node which is currently at the top border of the client area.

- **TotalCount**
  Returns the number of nodes in the tree.

- **TotalInternalDataSize**
Keeps the currently accumulated data size for one node.

- **TreeOptions**
  Reference to the tree's options.

- **TreeStates**
  Property which keeps a set of flags which indicate current operation and states of the tree.

- **UpdateCount**
  Not documented.

- **VerticalAlignment**
  Used to set a node's vertical button alignment with regard to the entire node rectangle.

- **VisibleCount**
  Number of currently visible nodes.

- **VisiblePath**
  Property to set or determine a node parent's expand states.

- **WantTabs**
  Read or set whether the tree wants to process tabs on its own.

### Events

- **OnGetHint**
  Virtual string tree event to query for a custom hint text.

- **OnGetText**
  Virtual string tree event to query for a node's normal or static text.

- **OnNewText**
  Virtual string tree event to pass edited text.

- **OnPaintText**
  Event to change text formatting for particular nodes.

- **OnShortenString**
  String tree event for custom handling of string abbreviations.

### TBaseVirtualTree Class

- **OnAdvancedHeaderDraw**
  Header paint support event.

- **OnAfterCellPaint**
Paint support event.

- OnAfterItemErase
  Paint support event.
- OnAfterItemPaint
  Paint support event.
- OnAfterPaint
  Paint support event.
- OnBeforeCellPaint
  Paint support event.
- OnBeforeItemErase
  Paint support event.
- OnBeforeItemPaint
  Paint support event.
- OnBeforePaint
  Paint support event.
- OnChange
  Navigation support event.
- OnChecked
  Check support event.
- OnChecking
  Check support event.
- OnCollapsed
  Miscellaneous event.
- OnCollapsing
  Miscellaneous event.
- OnColumnClick
  Header and column support event.
- OnColumnDblClick
  Header and column support event.
- OnColumnResize
  Header and column support routine.
- OnCompareNodes
  Sort and search support event.
- OnCreateDataObject
  Drag‘n drop support event.
**OnCreateDragManager**
Drag'n drop support event.

**OnCreateEditor**
Editing support event.

**OnDragAllowed**
Drag'n drop support event.

**OnDragDrop**
Drag'n drop support event.

**OnDragOver**
Drag'n drop support event.

**OnEditCancelled**
Editing support event.

**OnEdited**
Editing support event.

**OnEditing**
Editing support event.

**OnExpanded**
Misscellaneous event.

**OnExpanding**
Misscellaneous event.

**OnFocusChanged**
Navigation support event.

**OnFocusChanging**
Navigation support event.

**OnFreeNode**
Data management node.

**OnGetCellIsEmpty**
Triggered when the tree control needs to know whether a given column is empty.

**OnGetCursor**
Miscellaneous event.

**OnGetHeaderCursor**
Header and column support event.

**OnGetHelpContext**
Miscellaneous event.
- **OnGetImageIndex**
  Display management event.
- **OnGetImageIndexEx**
  Not documented.
- **OnGetLineStyle**
  Display management event.
- **OnGetNodeDataSize**
  Data management event.
- **OnGetPopupMenu**
  Miscellaneous event.
- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.
- **OnHeaderClick**
  Header & column support event.
- **OnHeaderDbClick**
  Header & column support event.
- **OnHeaderDragged**
  Header & column support event.
- **OnHeaderDraggedOut**
  Header & column support event.
- **OnHeaderDragging**
  Header & column support event.
- **OnHeaderDraw**
  Header & column support event.
- **OnHeaderDrawQueryElements**
  Header & column support event.
- **OnHeaderMouseDown**
  Header & column support event.
- **OnHeaderMouseMove**
  Header & column support event.
- **OnHeaderMouseUp**
  Header & column support event.
- **OnHotChange**
  Navigation support event.
- **OnIncrementalSearch**
Miscellaneous event.

 OnInitChildren  
 Node management event.

 OnInitNode  
 Node management event.

 OnKeyAction  
 Miscellaneous event.

 OnLoadNode  
 Streaming support event.

 OnMeasureItem  
 Miscellaneous event.

 OnNodeCopied  
 Miscellaneous event.

 OnNodeCopying  
 Miscellaneous event.

 OnNodeMoved  
 Miscellaneous event.

 OnNodeMoving  
 Miscellaneous event.

 OnPaintBackground  
 Paint support event.

 OnRenderOLEData  
 Drag'n drop and clipboard support event.

 OnResetNode  
 Node management event.

 OnSaveNode  
 Streaming support event.

 OnScroll  
 Miscellaneous event.

 OnShowScrollbar  
 Not documented.

 OnStateChange  
 Miscellaneous event.

 OnStructureChange  
 Miscellaneous event.
On Updating
Miscellaneous event.

Methods

- **AdjustPaintCellRect**
  Method which can be used by descendants to adjust the given rectangle during a paint cycle.

- **CalculateTextWidth**
  Not documented.

- **ColumnIsEmpty**
  Used to determine if a cell is considered as being empty.

- **ComputeNodeHeight**
  Not documented.

- **ContentToClipboard**
  Not documented.

- **ContentToHTML**
  Not documented.

- **ContentToRTF**
  Not documented.

- **ContentToText**
  Not documented.

- **ContentToUnicode**
  Not documented.

- **Create**
  Constructor of the control

- **DefineProperties**
  Helper method to customize loading and saving persistent tree data.

- **DoCreateEditor**
  Not documented.

- **DoGetNodeHint**
  Not documented.

- **DoGetNodeTooltip**
  Not documented.

- **DoGetNodeWidth**
Overridable method which always returns 0.

- `DoGetText`
  Not documented.
- `DoIncrementalSearch`
  Not documented.
- `DoNewText`
  Not documented.
- `DoPaintNode`
  Overridable method which does nothing.
- `DoPaintText`
  Not documented.
- `DoShortenString`
  Not documented.
- `DoTextDrawing`
  Not documented.
- `DoTextMeasuring`
  Not documented.
- `GetOptionsClass`
  Customization helper to determine which options class the tree should use.
- `GetTextInfo`
  Helper method for node editors, hints etc.
- `InternalData`
  Returns the address of the internal data for a tree class.
- `InvalidateNode`
  Invalidates the given node.
- `MainColumnChanged`
  Not documented.
- `Path`
  Not documented.
- `ReadChunk`
  Not documented.
- `ReadOldStringOptions`
  Not documented.
- `ReinitNode`
Forces a reinitialization of the given node.

- **RenderOLEData**
  Renders pending OLE data.

- **WriteChunks**
  Writes the core chunks for the given node to the given stream.

**TBaseVirtualTree Class**

- **AbsoluteIndex**
  Reads the overall index of a node.

- **AddChild**
  Creates and adds a new child node to given node.

- **AddFromStream**
  Adds the content from the given stream to the given node.

- **AddToSelection**
  Adds one or more nodes to the current selection.

- **AdjustPaintCellRect**
  Used in descendants to modify the clip rectangle of the current column while painting a certain node.

- **AdjustPanningCursor**
  Loads the proper cursor which indicates into which direction scrolling is done.

- **AdviseChangeEvent**
  Used to register a delayed change event.

- **AllocateInternalDataArea**
  Registration method to allocate tree internal data per node.

- **Animate**
  Support method for animated actions in the tree view.

- **Assign**
  Used to copy properties from another Virtual Treeview.

- **BeginDrag**
  Starts an OLE drag'n drop operation.

- **BeginSynch**
  Enters the tree into a special synchronized mode.

- **BeginUpdate**
  Enters the tree into a special synchronized mode.
Locks the tree view to perform several update operations.

- **CalculateSelectionRect**
  Support method for draw selection.

- **CanAutoScroll**
  Determines whether the tree can currently auto scroll its window.

- **CancelCutOrCopy**
  Cancels any pending cut or copy clipboard operation.

- **CancelEditNode**
  Cancel the current edit operation, if there is any.

- **CanEdit**
  Determines whether a node can be edited or not.

- **CanFocus**
  Support method to determine whether the tree window can receive the input focus.

- **CanShowDragImage**
  Determines whether a drag image should be shown.

- **Change**
  Central method called when a node's selection state changes.

- **ChangeScale**
  Helper method called by the VCL when control resizing is due.

- **CheckParentCheckState**
  Helper method for recursive check state changes.

- **Clear**
  Clears the tree and removes all nodes.

- **ClearChecked**
  Not documented.

- **ClearSelection**
  Removes all nodes from the current selection.

- **ClearTempCache**
  Helper method to clear the internal temporary node cache.

- **ColumnIsEmpty**
  Used to determine if a cell is considered as being empty.

- **CopyTo**
  Copies **Source** and all its child nodes to **Target**.

- **CopyToClipBoard**
Copies all currently selected nodes to the clipboard.

- **CountLevelDifference**
  Determines the level difference of two nodes.

- **CountVisibleChildren**
  Determines the number of visible child nodes of the given node.

- **Create**
  Constructor of the control

- **CreateParams**
  Prepares the creation of the controls window handle.

- **CreateWnd**
  Initializes data, which depends on the window handle.

- **CutToClipBoard**
  Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

- **DefineProperties**
  Helper method to customize loading and saving persistent tree data.

- **DeleteChildren**
  Removes all child nodes from the given node.

- **DeleteNode**
  Removes the given node from the tree.

- **DeleteSelectedNodes**
  Removes all currently selected nodes from the tree.

- **Destroy**
  Destructor of the control.

- **DetermineHiddenChildrenFlag**
  Determines whether all children of a given node are hidden.

- **DetermineHiddenChildrenFlagAllNodes**
  Determines whether all children of all nodes are hidden.

- **DetermineHitPositionLTR**
  Determines the hit position within a node with left-to-right and right-to-left orientation.

- **DetermineHitPositionRTL**
  Determines the hit position within a node with left-to-right and right-to-left orientation.

- **DetermineNextCheckState**
Not documented.

- **DetermineScrollDirections**
  Not documented.

- **DoAdvancedHeaderDraw**
  Not documented.

- **DoAfterCellPaint**
  Not documented.

- **DoAfterItemErase**
  Not documented.

- **DoAfterItemPaint**
  Not documented.

- **DoAfterPaint**
  Not documented.

- **DoAutoScroll**
  Enables or disables the auto scroll timer.

- **DoBeforeCellPaint**
  Not documented.

- **DoBeforeDrag**
  Not documented.

- **DoBeforeItemErase**
  Not documented.

- **DoBeforeItemPaint**
  Not documented.

- **DoBeforePaint**
  Not documented.

- **DoCancelEdit**
  Called when the tree should stop editing without accepting changed values.

- **DoCanEdit**
  Not documented.

- **DoChange**
  Not documented.

- **DoCheckClick**
  Not documented.

- **DoChecked**
Not documented.

DoChecking
Not documented.

DoCollapsed
Not documented.

DoCollapsing
Not documented.

DoColumnClick
Not documented.

DoColumnDblClick
Not documented.

DoColumnResize
Not documented.

DoCompare
Not documented.

DoCreateDataObject
Not documented.

DoCreateDragManager
Not documented.

DoCreateEditor
Not documented.

DoDragDrop
Not documented.

DoDragExpand
Not documented.

DoDragging
Internal method which handles drag' drop.

DoDragOver
Not documented.

DoEdit
Initiates editing of the currently set focused column and edit node.

DoEndDrag
Not documented.

DoEndEdit
Stops the current edit operation and takes over the new content.
DoExpanded
   Not documented.
DoExpanding
   Not documented.
DoFocusChange
   Not documented.
DoFocusChanging
   Not documented.
DoFocusNode
   Internal method to set the focused node.
DoFreeNode
   Not documented.
DoGetAnimationType
   Determines the type of animation to be used.
DoGetCursor
   Not documented.
DoGetHeaderCursor
   Not documented.
DoGetImageIndex
   Not documented.
DoGetLineStyle
   Not documented.
DoGetNodeHint
   Not documented.
DoGetNodeTooltip
   Not documented.
DoGetNodeWidth
   Overridable method which always returns 0.
DoGetPopupMenu
   Overridable method which triggers the OnGetPopup event.
DoGetUserClipboardFormats
   Not documented.
DoHeaderClick
   Not documented.
DoHeaderDblClick
Not documented.

- DoHeaderDragged
  Not documented.

- DoHeaderDraggedOut
  Not documented.

- DoHeaderDragging
  Not documented.

- DoHeaderDraw
  Not documented.

- DoHeaderDrawQueryElements
  Not documented.

- DoHeaderMouseDown
  Not documented.

- DoHeaderMouseMove
  Not documented.

- DoHeaderMouseUp
  Not documented.

- DoHotChange
  Not documented.

- DoIncrementalSearch
  Not documented.

- DoInitChildren
  Not documented.

- DoInitNode
  Not documented.

- DoKeyAction
  Not documented.

- DoLoadUserData
  Not documented.

- DoMeasureItem
  Not documented.

- DoNodeCopied
  Not documented.

- DoNodeCopying
  Not documented.
DoNodeMoved
Not documented.

DoNodeMoving
Not documented.

DoPaintBackground
Not documented.

DoPaintDropMark
Overridable method which draws the small line on top of a nodes image depending on the current drop state.

DoPaintNode
Overridable method which does nothing.

DoPopupMenu
Overridable method which shows the popup menu for the given node.

DoRenderOLEData
Not documented.

DoReset
Not documented.

DoSaveUserData
Not documented.

DoScroll
Overridable method which triggers the `OnScroll` event.

DoSetOffsetXY
Internal core routine to set the tree's scroll position.

DoShowScrollbar
Not documented.

DoStartDrag
Not documented.

DoStateChange
Not documented.

DoStructureChange
Not documented.

DoTimerScroll
Callback method which is triggered whenever the scroll timer fires.

DoUpdating
Not documented.
DoValidateCache
Not documented.

DragCanceled
Called by the VCL when a drag'n drop operation was canceled by the user.

DragDrop
Helper method, which is used when a drag operation is finished.

DragEnter
Not documented.

DragFinished
Called when a drag operation is finished (accepted or cancelled).

Dragging
Returns true if a drag'n drop operation is in progress.

DragLeave
Not documented.

DragOver
Not documented.

DrawDottedHLine
Not documented.

DrawDottedVLine
Not documented.

EditNode
Starts editing the given node if allowed to.

EndEditNode
Stops node editing if it was started before.

EndSynch
Counterpart to BeginSynch.

EndUpdate
Resets the update lock set by BeginUpdate.

ExecuteAction
Not documented.

FindNodeInSelection
Helper method to find the given node in the current selection.

FinishChunkHeader
Not documented.
FinishCutOrCopy
    Stops any pending cut or copy clipboard operation.
FlushClipboard
    Renders all pending clipboard data.
FontChanged
    Not documented.
FullCollapse
    Collapses all nodes in the tree.
FullExpand
    Expands all nodes in the tree.
GetBorderDimensions
    Not documented.
GetCheckImage
    Not documented.
GetCheckImageListFor
    Not documented.
GetColumnClass
    Returns the class to be used to manage columns in the tree.
GetControlsAlignment
    Not documented.
GetDisplayRect
    Returns the visible region used by the given node in client coordinates.
GetFirst
    Group of node navigation functions.
GetFirstChecked
    Not documented.
GetFirstChild
    Group of node navigation functions.
GetFirstCutCopy
    Group of node navigation functions.
GetFirstInitialized
    Group of node navigation functions.
GetFirstNoInit
    Group of node navigation functions.
GetFirstSelected
Group of node navigation functions.

GetFirstVisible
Group of node navigation functions.

GetFirstVisibleChild
Group of node navigation functions.

GetFirstVisibleChildNoInit
Group of node navigation functions.

GetFirstVisibleNoInit
Group of node navigation functions.

GetHeaderClass
Returns the header class to be used by the tree.

GetHintWindowClass
Not documented.

GetHitTestInfoAt
Returns information about the node at the given position.

GetImageIndex
Not documented.

GetLast
Group of node navigation functions.

GetLastChild
Group of node navigation functions.

GetLastChildNoInit
Group of node navigation functions.

GetLastInitialized
Group of node navigation functions.

GetLastNoInit
Group of node navigation functions.

GetLastVisible
Group of node navigation functions.

GetLastVisibleChild
Group of node navigation functions.

GetLastVisibleChildNoInit
Group of node navigation functions.

GetLastVisibleNoInit
Group of node navigation functions.

- **GetMaxColumnWidth**
  Returns the width of the largest node in the given column.

- **GetMaxRightExtend**
  Determines the maximum width of the currently visible part of the tree.

- **GetNativeClipboardFormats**
  Used to let descendants and the application add their own supported clipboard formats.

- **GetNext**
  Group of node navigation functions.

- **GetNextChecked**
  Not documented.

- **GetNextCutCopy**
  Group of node navigation functions.

- **GetNextInitialized**
  Group of node navigation functions.

- **GetNextNoInit**
  Group of node navigation functions.

- **GetNextSelected**
  Group of node navigation functions.

- **GetNextSibling**
  Group of node navigation functions.

- **GetNextVisible**
  Group of node navigation functions.

- **GetNextVisibleNoInit**
  Group of node navigation functions.

- **GetNextVisibleSibling**
  Group of node navigation functions.

- **GetNextVisibleSiblingNoInit**
  Group of node navigation functions.

- **GetNodeAt**
  Not documented.

- **GetNodeData**
  Returns the address of the user data area of the given node.

- **GetNodeLevel**
Returns the indentation level of the given node.

*GetOptionsClass*
Customization helper to determine which options class the tree should use.

*GetPrevious*
Group of node navigation functions.

*GetPreviousInitialized*
Group of node navigation functions.

*GetPreviousNoInit*
Group of node navigation functions.

*GetPreviousSibling*
Group of node navigation functions.

*GetPreviousVisible*
Group of node navigation functions.

*GetPreviousVisibleNoInit*
Group of node navigation functions.

*GetPreviousVisibleSibling*
Group of node navigation functions.

*GetPreviousVisibleSiblingNoInit*
Group of node navigation functions.

*GetSortedCutCopySet*
Returns a sorted list of nodes, which are marked for cut or copy clipboard operation.

*GetSortedSelection*
Returns a sorted list of all currently selected nodes.

*GetTextInfo*
Helper method for node editors, hints etc.

*GetTreeFromDataObject*
OLE drag'n drop and clipboard support method.

*GetSizeRect*
Returns the size of the virtual tree image.

*GetVisibleParent*
Returns the first (nearest) parent node, which is visible.

*HandleHotTrack*
Not documented.
HandleIncrementalSearch
Not documented.

HandleMouseDblClick
Not documented.

HandleMouseDown
Not documented.

HandleMouseUp
Not documented.

HasAsParent
Determines if the given node has got another node as one of its parents.

HasImage
Not documented.

HasPopupMenu
Determines whether there is a pop up menu assigned to the tree.

InitChildren
Not documented.

InitNode
Not documented.

InsertNode
Inserts a new node and returns it to the caller.

InternalAddFromStream
Not documented.

InternalAddToSelection
Not documented.

InternalCacheNode
Not documented.

InternalClearSelection
Not documented.

InternalConnectNode
Not documented.

InternalData
Returns the address of the internal data for a tree class.

InternalDisconnectNode
Not documented.
InternalRemoveFromSelection
Not documented.

InvalidateCache
Empties the internal node cache and marks it as invalid.

InvalidateChildren
Invalidates all children of the given node.

InvalidateColumn
Invalidates the client area part of a column.

InvalidateNode
Invalidates the given node.

InvalidateToBottom
Invalidates the client area starting with the top position of the given node.

InvertSelection
Inverts the current selection.

IsEditing
Tells the caller whether the tree is currently in edit mode.

IsMouseSelecting
Tell the caller whether the tree is currently in draw selection mode.

IterateSubtree
Iterator method to go through all nodes of a given sub tree.

Loaded
Not documented.

LoadFromFile
Loads previously streamed out tree data back in again.

LoadFromStream
Loads previously streamed out tree data back in again.

MainColumnChanged
Not documented.

MarkCutCopyNodes
Not documented.

MeasureItemHeight
Not documented.

MouseMove
Not documented.
MoveTo
Moves Source and all its child nodes to Target.

Notification
Not documented.

OriginalWMNCPaint
Not documented.

Paint
TControl's Paint method used here to display the tree.

PaintCheckImage
Not documented.

PaintImage
Not documented.

PaintNodeButton
Not documented.

PaintSelectionRectangle
Not documented.

PaintTree
Main paint routine for the tree image.

PaintTreeLines
Not documented.

PanningWindowProc
Not documented.

PasteFromClipboard
Inserts the content of the clipboard into the tree.

PrepareDragImage
Not documented.

Print
Not documented.

ProcessDrop
Helper method to ease OLE drag'n drop operations.

ProcessOLEData
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk
Not documented.

ReadNode
Not documented.

- `RedirectFontChangeEvent`
  Not documented.
- `ReinitChildren`
  Forces all child nodes of Node to be reinitialized.
- `ReinitNode`
  Forces a reinitialization of the given node.
- `RemoveFromSelection`
  Removes the given node from the current selection.
- `RenderOLEData`
  Renders pending OLE data.
- `RepaintNode`
  Causes the treeview to repaint the given node.
- `ResetNode`
  Resets the given node to uninitialized.
- `ResetRangeAnchor`
  Not documented.
- `RestoreFontChangeEvent`
  Not documented.
- `SaveToFile`
  Saves the entire content of the tree into a file or stream.
- `SaveToStream`
  Saves the entire content of the tree into a file or stream.
- `ScrollIntoView`
  Scrolls the tree so that the given node comes in the client area.
- `SelectAll`
  Selects all nodes in the tree.
- `SelectNodes`
  Selects a range of nodes.
- `SetBiDiMode`
  Not documented.
- `SetFocusedNodeAndColumn`
  Not documented.
- `SkipNode`
  Not documented.
Sort
Sorts the given node.

SortTree
Sorts the entire tree view.

StartWheelPanning
Not documented.

StopWheelPanning
Not documented.

StructureChange
Not documented.

SuggestDropEffect
Not documented.

ToggleNode
Changes a node's expand state to the opposite state.

ToggleSelection
Toggles the selection state of a range of nodes.

UnselectNodes
Deselects a range of nodes.

UpdateAction
Not documented.

UpdateDesigner
Not documented.

UpdateEditBounds
Not documented.

UpdateHeaderRect
Not documented.

UpdateHorizontalScrollBar
Applies changes to the horizontal and vertical scrollbars.

UpdateScrollBars
Applies changes to the horizontal and vertical scrollbars.

UpdateVerticalScrollBar
Applies changes to the horizontal and vertical scrollbars.

UpdateWindowAndDragImage
Not documented.

UseRightToLeftReading
Helper method for right-to-left layout.

- **ValidateCache**
  Initiates the validation of the internal node cache.

- **ValidateChildren**
  Validates all children of a given node.

- **ValidateNode**
  Validates a given node.

- **ValidateNodeDataSize**
  Helper method for node data size initialization.

- **WndProc**
  Redirected window procedure to do some special processing.

- **WriteChunks**
  Writes the core chunks for the given node to the given stream.

- **WriteNode**
  Writes the cover (envelop) chunk for the given node to the given stream.

### Legend

- protected
- Property
- read only
- public
- Event
- Method
- virtual

### Class Hierarchy

```
TCustomControl -> TBaseVirtualTree -> TCustomVirtualStringTree
```
**TCustomVirtualStringTree.DefaultText Property**

*TCustomVirtualStringTree Class*

Not documented.

**Pascal**

```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*

**Links**

*TCustomVirtualStringTree Class*

---

*What do you think about this topic? Send feedback!*
**TCustomVirtualStringTree.EllipsisWidth Property**

**TCustomVirtualStringTree Class**

Not documented.

**Pascal**

```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**

**Links**

**TCustomVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
**TCustomVirtualStringTree.OnGetHint Event**

Virtual string tree event to query for a custom hint text.

**Pascal**

```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**

TCustomVirtualStringTree Class

**Links**

TCustomVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
Virtual string tree event to query for a node's normal or static text.

Pascal

```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

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class, See Also
**TCustomVirtualStringTree.OnNewText Event**

Virtual string tree event to pass edited text.

**Pascal**

```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`
TCustomVirtualStringTree Class

Links
TCustomVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
**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` (when the conditions for the changed formatting are met - not in the event handler for `OnPaintText`).

**See Also**
- Paint cycles and stages

**Class**
- `TCustomVirtualStringTree Class`

**Links**
- `TCustomVirtualStringTree Class`, **See Also**

What do you think about this topic? Send feedback!
TCustomVirtualStringTree.OnShortenString Event

String tree event for custom handling of string abbreviations.

Pascal

```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
TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
TCustomVirtualStringTree.Text Property

TCustomVirtualStringTree Class

Not documented.

Pascal

```pascal
property Text [Node: PVirtualNode; Column: TColumnInfo]
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
**TCustomVirtualStringTree.TreeOptions Property**

**TCustomVirtualStringTree Class**

Reference to the tree's options.

**Pascal**

```pascal
property TreeOptions: TCustomStringTreeOptions;
```

**Description**

The tree options are one of the main switches to modify a treeview's behavior. Virtual Treeview supports customizing tree options by descendants. 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 descendants.

**Class**

TCustomVirtualStringTree Class

**Links**

TCustomVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TCustomVirtualStringTree.AdjustPaintCellRect Method**

TCustomVirtualStringTree Class

Method which can be used by descendents to adjust the given rectangle during a paint cycle.

**Pascal**

```
procedure AdjustPaintCellRect(var PaintInfo: TVTPaintInfo);
```

**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

**Links**

TCustomVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TCustomVirtualStringTree.CalculateTextWidth Method**

**TCustomVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
function CalculateTextWidth(Canvas: TCanvas; Node: PP
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TCustomVirtualStringTree Class

**Links**

TCustomVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TCustomVirtualStringTree.ColumnIsEmpty** Method

*See Also*  
Used to determine if a cell is considered as being empty.

**Pascal**

```pascal
function ColumnIsEmpty(Node: PVirtualNode; Column: TColumnIndex): Boolean;
```

**Description**

An empty cell might be used for the automatic column spanning feature. Descendants can override this method to modify the tree's behavior.

**See Also**

toAutoSpanColumns

**Class**

*TCustomVirtualStringTree Class*

**Links**

*TCustomVirtualStringTree Class, See Also*

*What do you think about this topic? Send feedback!*
**TCustomVirtualStringTree.ComputeNodeHeight**

**Method**

**TCustomVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
function ComputeNodeHeight(Canvas: TCanvas; Node: PV)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TCustomVirtualStringTree Class**

**Links**

**TCustomVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
TCustomVirtualStringTree.ContentToClipboard

### Method

**TCustomVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
function ContentToClipboard(Format: Word; Source: TVSTTextSourceType): string;
function ContentToHTML(Source: TVSTTextSourceType; Caption: WideString = ''): WideString;
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**

### Links

**TCustomVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
TCustomVirtualStringTree.Create Constructor

Constructors of the control

Pascal

```
constructor Create(AOwner: TComponent); override;
```

Description

The constructor initializes certain properties to their default values.

Class

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class

---

What do you think about this topic? Send feedback!
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

Links

TCustomVirtualStringTree Class
What do you think about this topic? Send feedback!
TCustomVirtualStringTree.DoCreateEditor Method
TCustomVirtualStringTree Class

Not documented.

Pascal

```
function DoCreateEditor(Node: PVirtualNode; Column:

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TCustomVirtualStringTree Class

Links
TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
**TCustomVirtualStringTree.DoGetNodeHint** Method

**TCustomVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
function DoGetNodeHint(Node: PVirtualNode; Column: TColumnIndex) : string;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TCustomVirtualStringTree Class**

**Links**

**TCustomVirtualStringTree Class**

---

*What do you think about this topic? Send feedback!*
**TCustomVirtualStringTree.DoGetNodeTooltip Method**

**TCustomVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
function DoGetNodeTooltip(Node: PVirtualNode; Column
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TCustomVirtualStringTree Class**

**Links**

**TCustomVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
**TCustomVirtualStringTree.DoGetNodeWidth Method**

**TCustomVirtualStringTree Class**

Overridable method which always returns 0.

**Pascal**

```pascal
function DoGetNodeWidth(Node: PVirtualNode; Column: TColumnIndex): Integer;
```

**Description**

Descendants 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

**Links**

TCustomVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TCustomVirtualStringTree.DoGetText Method

Pascal

```
procedure DoGetText(Node: PVirtualNode; Column: TColumnIndex);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
### TCustomVirtualStringTree.DoIncrementalSearch Method

**TCustomVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
function DoIncrementalSearch(Node: PVirtualNode; const Description: string): boolean;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TCustomVirtualStringTree Class

**Links**

TCustomVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
TCustomVirtualStringTree.DoNewText Method

TCustomVirtualStringTree Class

Not documented.

Pascal

```pascal
procedure DoNewText(Node: PVirtualNode; Column: TColumn);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
**TCustomVirtualStringTree.DoPaintNode**

**Method**

**TCustomVirtualStringTree Class**

Overridable method which does nothing.

**Pascal**

```pascal
procedure DoPaintNode(var PaintInfo: TVTPaintInfo);
```

**Description**

Descendants override this method to paint the content of the node. For instance string trees draw the node's caption.

**Class**

TCustomVirtualStringTree Class

**Links**

TCustomVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
TCustomVirtualStringTree.DoPaintText Method

TCustomVirtualStringTree Class

Not documented.

Pascal

```pascal
procedure DoPaintText(Node: PVirtualNode; const Canvas: TCanvas);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
TCustomVirtualStringTree.DoShortenString Method

TCustomVirtualStringTree Class

Not documented.

Pascal

```
function DoShortenString(Canvas: TCanvas; Node: PVirt;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TCustomVirtualStringTree.DoTextDrawing Method

TCustomVirtualStringTree Class

Not documented.

Pascal

```
procedure DoTextDrawing(var PaintInfo: TVTPaintInfo);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
**TCustomVirtualStringTree.DoTextMeasuring**

**Method**

TCustomVirtualStringTree Class

Not documented.

**Pascal**

```pascal
function DoTextMeasuring(Canvas: TCanvas; Node: PVirtualNode): Integer;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TCustomVirtualStringTree Class

**Links**

TCustomVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TCustomVirtualStringTree.GetOptionsClass Method

Customization helper to determine which options class the tree should use.

Pascal

```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:
```pascal
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;
  ...
end;

function TVirtualStringTree.GetOptions: TStringTreeOptions;
begin
  Result := FOptions as TStringTreeOptions;
end;

procedure TVirtualStringTree.SetOptions(const Value: TArrayTreeOptions);
begin
  FOptions.Assign(Value);
end;

function TVirtualStringTree.GetOptionsClass: TTreeOptionsClass;
```

begin
  Result := TStringTreeOptions;
end;

Class
  TCustomVirtualStringTree Class

Links
  TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
**TCustomVirtualStringTree.GetTextInfo Method**

**TCustomVirtualStringTree Class**

Helper method for node editors, hints etc.

**Pascal**

```pascal
procedure GetTextInfo(Node: PVirtualNode; Column: TColumnIndex);
```

**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

**Links**

TCustomVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TCustomVirtualStringTree.InternalData Method**

**TCustomVirtualStringTree Class | See Also**

Returns the address of the internal data for a tree class.

**Pascal**

```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

**Class**

TCustomVirtualStringTree Class

**Links**

TCustomVirtualStringTree Class, See Also

*What do you think about this topic? Send feedback!*
TCustomVirtualStringTree.InvalidateNode Method

TCustomVirtualStringTree Class

InvalidateNode method invalidates the given node.

Pascal

```
function InvalidateNode(Node: PVirtualNode): TRect;
```

Description

InvalidateNode method initiates repaint of the given node by calling InvalidateRect with the node's display rectangle and returns this rectangle.

Class

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
**TCustomVirtualStringTree.MainColumnChanged Method**

**TCustomVirtualStringTree Class**

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

**Links**

TCustomVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TCustomVirtualStringTree.Path Method**

**TCustomVirtualStringTree Class**

Not documented.

Pascal

```pascal
function Path(Node: PVirtualNode; Column: TColumnIndex): String;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TCustomVirtualStringTree Class

**Links**

TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
TCustomVirtualStringTree.ReadChunk Method

TCustomVirtualStringTree Class

Not documented.

Pascal

```pascal
function ReadChunk(Stream: TStream; Version: Integer)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
TCustomVirtualStringTree.ReadOldStringOptions Method

TCustomVirtualStringTree Class

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

Links

TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
**TCustomVirtualStringTree.ReinitNode Method**

**TCustomVirtualStringTree Class**

Forces a reinitialization of the given node.

**Pascal**

```pascal
procedure ReinitNode(Node: PVirtualNode; Recursive: Boolean);
```

**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**

**Links**

**TCustomVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
TCustomVirtualStringTree.RenderOLEData Method

TCustomVirtualStringTree Class

Renders pending OLE data.

Pascal

function RenderOLEData(const FormatEtcIn: TFormatEtc) ...

Description

RenderOLEData 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

Links
TCustomVirtualStringTree Class

What do you think about this topic? Send feedback!
TCustomVirtualStringTree.WriteChunks Method
TCustomVirtualStringTree Class | See Also

Writes the core chunks for the given node to the given stream.

Pascal

```pascal
procedure WriteChunks(Stream: TStream; Node: PVirtualNode);
```

Description

WriteChunks is part of the streaming system in Virtual Treeview and writes the core chunks for `Node` into `Stream`. Descendants 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

Links
TCustomVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TCustomVirtualTreeOptions Class

Classes | Methods | Properties | Legend

Organizes all tree options into subproperties for easier management.

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

Members

Properties

- **AnimationOptions**
  Options related to animations.
- **AutoOptions**
  Options related to automatic actions.
- **MiscOptions**
Options not related to any other category.

- **Owner**
  Owner tree to which the property class belongs.

- **PaintOptions**
  Options related to painting.

- **SelectionOptions**
  Options related to the way nodes can be selected.

**Methods**

- **AssignTo**
  Used to copy this option class to another option collection.

- **Create**
  Constructor of the class.

**Legend**

- protected
- Property
- public
- read only
- Method
- virtual

**Class Hierarchy**

```
TPersistent  ➔ TCustomVirtualTreeOptions
```

**File**

VirtualTrees

**Links**

Classes, Methods, Properties, Legend
What do you think about this topic? Send feedback!
**TCustomVirtualTreeOptions.AnimationOptions Property**

Options related to animations.

**Pascal**

```pascal
property AnimationOptions: TVTAnimationOptions;
```

**Description**

These options can be used to switch certain animation effects in a tree.

**Class**

TCustomVirtualTreeOptions Class

**Links**

TCustomVirtualTreeOptions Class

---

*What do you think about this topic? Send feedback!*
**TCustomVirtualTreeOptions.AutoOptions Property**

Options related to automatic actions.

**Pascal**

```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

**Links**

TCustomVirtualTreeOptions Class
TCustomVirtualTreeOptions.MiscOptions Property
TCustomVirtualTreeOptions Class

Options not related to any other category.

Pascal

```pascal
property MiscOptions: TVTMiscOptions;
```

Description
These options can be used to switch miscellaneous aspects in a tree.

Class
TCustomVirtualTreeOptions Class

Links
TCustomVirtualTreeOptions Class

What do you think about this topic? Send feedback!
## TCustomVirtualTreeOptions.Owner Property

### Description

Owner tree to which the property class belongs.

### Class

TCustomVirtualTreeOptions Class

### Links

TCustomVirtualTreeOptions Class

---

What do you think about this topic? Send feedback!
**TCustomVirtualTreeOptions.PaintOptions**

*TCustomVirtualTreeOptions Class*

Options related to painting.

**Pascal**

```
property PaintOptions: TVTPaintOptions;
```

**Description**

These options can be used to switch visual aspects of a tree.

**Class**

*TCustomVirtualTreeOptions Class*

**Links**

*TCustomVirtualTreeOptions Class*

*What do you think about this topic? Send feedback!*
**TCustomVirtualTreeOptions.SelectionOptions** Property

*TCustomVirtualTreeOptions Class*

Options related to the way nodes can be selected.

**Pascal**

```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*

**Links**

*TCustomVirtualTreeOptions Class*

*What do you think about this topic? Send feedback!*
**TCustomVirtualTreeOptions.AssignTo Method**

*TCustomVirtualTreeOptions Class*

Used to copy this option class to another option collection.

**Pascal**

```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*

**Links**

*TCustomVirtualTreeOptions Class*
**TCustomVirtualTreeOptions.Create**

**Constructor**

Constructor of the class.

**Pascal**

```pascal
constructor Create(AOwner: TBaseVirtualTree); virtual
```

**Description**

Used to assign default values to all sub lists.

**Class**

*TCustomVirtualTreeOptions Class*

**Links**

*TCustomVirtualTreeOptions Class*
**TEnumFormatEtc Class**

**Classes | Methods | Legend**

```plaintext
TEnumFormatEtc = class(TInterfacedObject, IEnumFormatEtc);
```

**Group**  
**Classes**

**Members**

**Methods**

- **Clone**  
  Not documented.
- **Create**  
  Not documented.
- **Next**  
  Not documented.
- **Reset**  
  Not documented.
- **Skip**  
  Not documented.

**Legend**

- `public`
- `Method`

**Class Hierarchy**

```
TInterfacedObject
   `--- TEnumFormatEtc
       `--- IEnumFormatEtc
```
What do you think about this topic? Send feedback!
**TEnumFormatEtc.Clone Method**

**TEnumFormatEtc Class**

Not documented.

**Pascal**

```pascal
function Clone(out Enum: IEnumFormatEtc): HRESULT;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TEnumFormatEtc Class

**Links**

TEnumFormatEtc Class

---

*What do you think about this topic? Send feedback!*
TEnumFormatEtc.Create Constructor

TEnumFormatEtc Class

Not documented.

Pascal

```
constructor Create(Tree: TBaseVirtualTree; AFormatEtcArray: TEnumFormatEtcList);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TEnumFormatEtc Class

Links

TEnumFormatEtc Class

What do you think about this topic? Send feedback!
**TEnumFormatEtc.Next Method**

**TEnumFormatEtc Class**

Not documented.

**Pascal**

```pascal
function Next(celt: Integer; out elt; pceltFetched: )
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TEnumFormatEtc Class

**Links**

TEnumFormatEtc Class

*What do you think about this topic? Send feedback!*
TEnumFormatEtc.Reset Method

TEnumFormatEtc Class

Not documented.

Pascal

```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

Links

TEnumFormatEtc Class

What do you think about this topic? Send feedback!
TEnumFormatEtc.Skip Method

TEnumFormatEtc Class

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

Links
TEnumFormatEtc Class

What do you think about this topic? Send feedback!
# TScrollBarOptions Class

```pascal
tScrollBarOptions = class(TPersistent);
```

## Group
- Classes

## Members
### Properties
- `AlwaysVisible`
  - Not documented.
- `HorizontalIncrement`
  - Not documented.
- `ScrollBars`
  - Not documented.
- `ScrollBarStyle`
  - Not documented.
- `VerticalIncrement`
  - Not documented.

### Methods
- `Assign`
  - Not documented.
- `Create`
  - Not documented.
- `GetOwner`
  - Not documented.
TScrollBarOptions.AlwaysVisible Property
TScrollBarOptions Class

Not documented.

Pascal

```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

Links
TScrollBarOptions Class

What do you think about this topic? Send feedback!
**TScrollBarOptions.HorizontalIncrement Property**

*Not documented.*

**Pascal**

```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*

**Links**

*TScrollBarOptions Class*
TScrollBarOptions Class

TScrollBarOptions.HorizontalIncrement Property

TScrollBarOptions.ScrollBarStyle Property

TScrollBarOptions.ScrollBars Property

TScrollBarOptions Class

Not documented.

Pascal

```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

Links

TScrollBarOptions Class

What do you think about this topic? Send feedback!
**TScrollBarOptions.ScrollBarStyle**

**Property**

*TScrollBarOptions Class*

Not documented.

**Pascal**

```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*

**Links**

*TScrollBarOptions Class*

*What do you think about this topic? Send feedback!*
**TScrollBarOptions.VerticalIncrement**

**Property**

TScrollBarOptions Class

Not documented.

Pascal

```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

**Links**

TScrollBarOptions Class

---

*What do you think about this topic? Send feedback!*
TScrollBarOptions.Assign Method
TScrollBarOptions Class

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

Links

TScrollBarOptions Class

What do you think about this topic? Send feedback!
TScrollBarOptions.Create Constructor

TScrollBarOptions Class

Not documented.

Pascal

```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

Links
TScrollBarOptions Class

What do you think about this topic? Send feedback!
TScrollBarOptions.GetOwner Method
TScrollBarOptions Class

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

Links
TScrollBarOptions Class

What do you think about this topic? Send feedback!
TStringEditLink Class

Classes | Methods | Properties | Legend

TStringEditLink is the standard node editor of a TVirtualStringTree.

Pascal

TStringEditLink = class(TInterfacedObject, IVTEditLink)

Description

TStringEditLink implements the interface IVTEditLink. 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 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

Members

Properties

Edit
Not documented.

Methods

BeginEdit
This function will be called by the virtual string tree when the editing starts.

CancelEdit
This function will be called by the virtual string tree when the current editing is about to be cancelled.

Create
Constructor of the class.

Destroy
Destructor of the class.

EndEdit
This function will be called by the virtual string tree when the current editing is being finished.

GetBounds
The virtual string tree uses this function to get the current bounding rect of the node editor.

PrepareEdit
This function is called by a virtual string tree to initialize the node editor.
ProcessMessage
This function is used to forward messages being directed to the virtual string tree.

SetBounds
The virtual string tree calls this function to initialize the bounding rect of the node editor.

IVTEditLink Interface

BeginEdit
This function will be called by the virtual tree when the editing starts.

CancelEdit
This function will be called by the virtual tree when the current editing is about to be cancelled.

EndEdit
This function will be called by the virtual tree when the current editing is being finished.

GetBounds
The virtual tree can use this function to get the current bounding rect of the node editor.

PrepareEdit
This function is called by a virtual tree to initialize the node editor.

ProcessMessage
This function is used to forward messages being directed to the virtual tree.

SetBounds
The virtual tree calls this function to initialize the bounding rectangle of the node editor.

Legend

- public
- Property
- Method
virtual

Class Hierarchy

File

VirtualTrees

Links

Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TStringEditLink(Edit Property)

Not documented.

Pascal

```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

Links

TStringEditLink Class

What do you think about this topic? Send feedback!
TStringEditLink.BeginEdit Method

TStringEditLink Class

This function will be called by the virtual string tree when the editing starts.

Pascal

```pascal
function BeginEdit: Boolean; virtual; stdcall;
```

Description

Please see interface IVTEditLink for a detailed explanation of this interface function.

Class

TStringEditLink Class

Links

TStringEditLink Class

What do you think about this topic? Send feedback!
**TStringEditLink.CancelEdit Method**

*TStringEditLink Class*

This function will be called by the virtual string tree when the current editing is about to be cancelled.

**Pascal**

```pascal
function CancelEdit: Boolean; virtual; stdcall;
```

**Description**

Please see interface *IVTEditLink* for a detailed explanation of this interface function.

**Class**

*TStringEditLink Class*

**Links**

*TStringEditLink Class*

*What do you think about this topic? Send feedback!*
TStringEditLink.Create Constructor
TStringEditLink Class

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

Links
TStringEditLink Class

What do you think about this topic? Send feedback!
**TStringEditLink.Destroy Destructor**

**TStringEditLink Class**

Destructor of the class.

**Pascal**

```pascal
destructor Destroy; override;
```

**Description**

Frees the internal editor control.

**Class**

TStringEditLink Class

**Links**

TStringEditLink Class

---

*What do you think about this topic? Send feedback!*
**TStringEditLink.EndEdit Method**

**TStringEditLink Class**

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 for a detailed explanation of this interface function.

**Class**

**TStringEditLink Class**

**Links**

**TStringEditLink Class**

*What do you think about this topic? Send feedback!*
**TStringEditLink.GetBounds Method**

*TStringEditLink Class*

The virtual string tree uses this function to get the current bounding rect of the node editor.

**Pascal**

```pascal
function GetBounds: TRect; virtual; stdcall;
```

**Description**

Please see interface **IVTEditLink** for a detailed explanation of this interface function.

**Class**

*TStringEditLink Class*

**Links**

*TStringEditLink Class*

What do you think about this topic? Send feedback!
TStringEditLink::PrepareEdit Method

This function is called by a virtual string tree to initialize the node editor.

Pascal

```pascal
function PrepareEdit(Tree: TBaseVirtualTree; Node: PW
```n

Description

Please see interface IVTEditLink for a detailed explanation of this interface function.

Class

TStringEditLink Class

Links

TStringEditLink Class
TStringEditLink.ProcessMessage Method

This function is used to forward messages being directed to the virtual string tree.

Pascal

```pascal
procedure ProcessMessage(var Message: TMessage); virtual
```

Description

Please see interface IVTEditLink for a detailed explanation of this interface function.

Class

TStringEditLink Class

Links

TStringEditLink Class

What do you think about this topic? Send feedback!
TStringEditLink.SetBounds Method

TStringEditLink Class

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 for a detailed explanation of this interface function.

Class

TStringEditLink Class

Links

TStringEditLink Class

What do you think about this topic? Send feedback!
TStringTreeOptions Class

Options class used in the string tree and its descendants.

Pascal

```
TStringTreeOptions = class(TCustomStringTreeOptions)
```

Description
This options class publishes all properties inherited from its ancestor and does not add any further functionality.

Group
Classes

Members

Properties

- **AnimationOptions**
  Options related to animations.
- **AutoOptions**
  Options related to automatic actions.
- **MiscOptions**
  Options not related to any other category.
- **PaintOptions**
  Options related to painting.
- **SelectionOptions**
  Options related to the way nodes can be selected.
- **StringOptions**
  The new options introduced by the class.
**TCustomStringTreeOptions Class**

- **StringOptions**
  The new options introduced by the class.

**TCustomVirtualTreeOptions Class**

- **AnimationOptions**
  Options related to animations.
- **AutoOptions**
  Options related to automatic actions.
- **MiscOptions**
  Options not related to any other category.
- **Owner**
  Owner tree to which the property class belongs.
- **PaintOptions**
  Options related to painting.
- **SelectionOptions**
  Options related to the way nodes can be selected.

**Methods**

**TCustomStringTreeOptions Class**

- **AssignTo**
  Used to copy the options class.
- **Create**
  The constructor of the class.

**TCustomVirtualTreeOptions Class**

- **AssignTo**
  Used to copy this option class to another option collection.
- **Create**
  Constructor of the class.

**Legend**
published
Property
protected
public
read only
Method
virtual

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TStringTreeOptions.AnimationOptions Property

TStringTreeOptions Class

Options related to animations.

Pascal

```pascal
property AnimationOptions: TVTAnimationOptions;
```

Description

These options can be used to switch certain animation effects in a tree.

Class

TStringTreeOptions Class

Links

TStringTreeOptions Class

What do you think about this topic? Send feedback!
**TStringTreeOptions.AutoOptions Property**

**TStringTreeOptions Class**

Options related to automatic actions.

**Pascal**

```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**

**Links**

**TStringTreeOptions Class**

*What do you think about this topic? Send feedback!*
**TStringTreeOptions.MiscOptions Property**

Options not related to any other category.

**Pascal**

```pascal
property MiscOptions: TVTMiscOptions;
```

**Description**

These options can be used to switch miscellaneous aspects in a tree.

**Class**

*TStringTreeOptions Class*

**Links**

*TStringTreeOptions Class*
**TStringTreeOptions.PaintOptions Property**

**TStringTreeOptions Class**

Options related to painting.

**Pascal**

```
property PaintOptions: TVTPaintOptions;
```

**Description**

These options can be used to switch visual aspects of a tree.

**Class**

TStringTreeOptions Class

**Links**

TStringTreeOptions Class

---

What do you think about this topic? Send feedback!
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

Links

TStringTreeOptions Class

What do you think about this topic? Send feedback!
TStringTreeOptions.StringOptions Property

The new options introduced by the class.

Pascal

```pascal
property StringOptions: TVTStringOptions;
```

Description

StringOptions provides access to the newly introduced options by which the base class is extended.

Class

TStringTreeOptions Class

Links

TStringTreeOptions Class

What do you think about this topic? Send feedback!
TVirtualDrawTree Class

Descendant of TBaseVirtualTree, 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.

Group

Classes

Members

Properties

- **Action**
  Not documented.
- **Align**
  Not documented.
- **Alignment**
  Determines the horizontal alignment of text if no columns are defined.
- **Anchors**
  Not documented.
- **AnimationDuration**
  Determines the maximum duration the tree can use to play an animation.
AutoExpandDelay
Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay
Time which determines when auto scrolling should start.

AutoScrollInterval
Time interval between scroll events when doing auto scroll.

Background
Holds a background image for the tree.

BackgroundOffsetX
Horizontal offset of the background image.

BackgroundOffsetY
Vertical offset of the background image.

BevelEdges
Not documented.

BevelInner
Not documented.

BevelKind
Not documented.

BevelOuter
Not documented.

BevelWidth
Not documented.

BiDiMode
Not documented.

BorderStyle
Same as TForm.BorderStyle.

BorderWidth
Not documented.

ButtonFillMode
Determines how to fill the background of the node buttons.

ButtonText
Determines the look of node buttons.

Canvas
Not documented.
ChangeDelay
Time which determines when the OnChange event should be triggered after the actual change event.

CheckImageKind
Determines which images should be used for checkboxes and radio buttons.

ClipboardFormats
Special class to keep a list of clipboard format descriptions.

Color
Not documented.

Colors
A collection of colors used in the tree.

Constraints
Not documented.

Ctl3D
Not documented.

CustomCheckImages
Assign your own image list to get the check images you like most.

DefaultNodeHeight
Read or set the height new nodes get as initial value.

DefaultPasteMode
Read or set the value, which determines where to add pasted nodes to.

DragCursor
Not documented.

DragHeight
Read or set the vertical limit of the internal drag image.

DragImageKind
Read or set what should be shown in the drag image.

DragKind
Not documented.

DragMode
Not documented.

DragOperations
Read or set which drag operations may be allowed in the tree.
DragType
Read or set which subsystem should be used for dragging.

DragWidth
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode
Read or set how multiselection with the mouse is to be visualized.

EditDelay
Read or set the maximum time between two single clicks on the same node, which should start node editing.

Enabled
Not documented.

Font
Same as TWinControl.Font.

Header
Provides access to the header instance.

HintAnimation
Read or set the current hint animation type.

HintMode
Read or set what type of hint you want for the tree view.

HotCursor
Read or set which cursor should be used for hot nodes.

Images
Read or set the tree's normal image list.

IncrementalSearch
Read or set the current incremental search mode.

IncrementalSearchDirection
Read or set the direction to be used for incremental search.

IncrementalSearchStart
Read or set where to start incremental search.

IncrementalSearchTimeout
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Indent
Read or set the indentation amount for node levels.

LineMode
Read or set the mode of the tree lines.

- **LineStyle**
  Read or set the mode of the tree lines.

- **Margin**
  Read or set the tree's node margin.

- **NodeAlignment**
  Read or set the node alignment value.

- **NodeDataSize**
  Read or set the extra data size for each node.

- **OnClick**
  Not documented.

- **OnDbClick**
  Not documented.

- **OnEndDock**
  Not documented.

- **OnEndDrag**
  Not documented.

- **OnEnter**
  Not documented.

- **OnExit**
  Not documented.

- **OnKeyDown**
  Not documented.

- **OnKeyPress**
  Not documented.

- **OnKeyUp**
  Not documented.

- **OnMouseDown**
  Not documented.

- **OnMouseMove**
  Not documented.

- **OnMouseUp**
  Not documented.

- **OnMouseWheel**
  Not documented.
- **OnResize**
  Not documented.

- **OnStartDock**
  Not documented.

- **ParentBiDiMode**
  Not documented.

- **ParentColor**
  Not documented.

- **ParentCtl3D**
  Not documented.

- **ParentFont**
  Not documented.

- **ParentShowHint**
  Not documented.

- **PopupMenu**
  Not documented.

- **RootNodeCount**
  Read or set the number of nodes on the top level.

- **ScrollBarOptions**
  Reference to the scroll bar options class.

- **SelectionBlendFactor**
  Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

- **SelectionCurveRadius**
  Read or set the current corner radius for node selection rectangles.

- **ShowHint**
  Not documented.

- **StateImages**
  Reference to the images list which is used for the state images.

- **TabOrder**
  Not documented.

- **TabStop**
  Not documented.

- **TextMargin**
  Read or set the distance of the node caption to its borders.
**TreeOptions**
Reference to the tree's options.

**Visible**
Not documented.

**WantTabs**
Read or set whether the tree wants to process tabs on its own.

**TBaseVirtualTree Class**

**Alignment**
Determines the horizontal alignment of text if no columns are defined.

**AnimationDuration**
Determines the maximum duration the tree can use to play an animation.

**AutoExpandDelay**
Time delay after which a node gets expanded if it is the current drop target.

**AutoScrollDelay**
Time which determines when auto scrolling should start.

**AutoScrollInterval**
Time interval between scroll events when doing auto scroll.

**Background**
Holds a background image for the tree.

**BackgroundOffsetX**
Horizontal offset of the background image.

**BackgroundOffsetY**
Vertical offset of the background image.

**BorderStyle**
Same as TForm.BorderStyle.

**ButtonFillMode**
Determines how to fill the background of the node buttons.

**ButtonStyle**
Determines the look of node buttons.

**ChangeDelay**
Time which determines when theOnChange event should be triggered after the actual change event.
CheckImageKind
Determines which images should be used for checkboxes and radio buttons.

CheckImages
Not documented.

CheckState
Read or set the check state of a node.

CheckType
Read or set the check type of a node.

ChildCount
Read or set the number of child nodes of a node.

ChildrenInitialized
Read whether a node's child count has been initialized already.

ClipboardFormats
Special class to keep a list of clipboard format descriptions.

Colors
A collection of colors used in the tree.

CustomCheckImages
Assign your own image list to get the check images you like most.

DefaultNodeHeight
Read or set the height new nodes get as initial value.

DefaultPasteMode
Read or set the value, which determines where to add pasted nodes to.

DragHeight
Read or set the vertical limit of the internal drag image.

DragImage
Holds the instance of the internal drag image.

DragImageKind
Read or set what should be shown in the drag image.

DragManager
Holds the reference to the internal drag manager.

DragOperations
Read or set which drag operations may be allowed in the tree.

DragSelection
Keeps a temporary list of nodes during drag'n drop.

- **DragType**
  Read or set which subsystem should be used for dragging.

- **DragWidth**
  Read or set the horizontal limit of the internal drag image.

- **DrawSelectionMode**
  Read or set how multiselection with the mouse is to be visualized.

- **DropTargetNode**
  Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

- **EditColumn**
  Not documented.

- **EditDelay**
  Read or set the maximum time between two single clicks on the same node, which should start node editing.

- **EditLink**
  Keeps a reference to the internal edit link during a node edit operation.

- **Expanded**
  Read or set the expanded state of a particular node.

- **FocusedColumn**
  Read or set the currently focused column.

- **FocusedNode**
  Read or set the currently focused node.

- **Font**
  Same as TWinControl.Font.

- **FullyVisible**
  Read or set whether a node is fully visible or not.

- **HasChildren**
  Read or set whether a node has got children.

- **Header**
  Provides access to the header instance.

- **HeaderRect**
  Returns the non-client-area rectangle used for the header.

- **HintAnimation**
Read or set the current hint animation type.
- **HintMode**
Read or set what type of hint you want for the tree view.
- **HotCursor**
Read or set which cursor should be used for hot nodes.
- **HotNode**
Read, which node is currently the hot node.
- **Images**
Read or set the tree's normal image list.
- **IncrementalSearch**
Read or set the current incremental search mode.
- **IncrementalSearchDirection**
Read or set the direction to be used for incremental search.
- **IncrementalSearchStart**
Read or set where to start incremental search.
- **IncrementalSearchTimeout**
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.
- **Indent**
Read or set the indentation amount for node levels.
- **IsDisabled**
Read or set the enabled state of the given node.
- **IsVisible**
Read or set the visibility state of the given node.
- **LastClickPos**
Used for retained drag start and wheel mouse scrolling.
- **LastDropMode**
Read how the last drop operation finished.
- **LineMode**
Read or set the mode of the tree lines.
- **LineStyle**
Read or set the mode of the tree lines.
- **Margin**
Read or set the tree's node margin.
- **MultiLine**
Read or toggle the multiline feature for a given node.

- **NodeAlignment**
  Read or set the node alignment value.

- **NodeDataSize**
  Read or set the extra data size for each node.

- **NodeHeight**
  Read or set a node's height.

- **NodeParent**
  Read or set a node's parent node.

- **OffsetX**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetXY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **RootNode**
  Reference to the internal root node which is the anchor of the entire tree node hierarchy.

- **RootNodeCount**
  Read or set the number of nodes on the top level.

- **ScrollBarOptions**
  Reference to the scroll bar options class.

- **SearchBuffer**
  Current input string for incremental search.

- **Selected**
  Property to modify or determine the selection state of a node.

- **SelectedCount**
  Contains the number of selected nodes.

- **SelectionBlendFactor**
  Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

- **SelectionCurveRadius**
  Read or set the current corner radius for node selection rectangles.

- **StateImages**
  Reference to the images list which is used for the state images.
- **TextMargin**
  Read or set the distance of the node caption to its borders.

- **TopNode**
  The top node is the node which is currently at the top border of the client area.

- **TotalCount**
  Returns the number of nodes in the tree.

- **TotalInternalDataSize**
  Keeps the currently accumulated data size for one node.

- **TreeOptions**
  Reference to the tree's options.

- **TreeStates**
  Property which keeps a set of flags which indicate current operation and states of the tree.

- **UpdateCount**
  Not documented.

- **VerticalAlignment**
  Used to set a node's vertical button alignment with regard to the entire node rectangle.

- **VisibleCount**
  Number of currently visible nodes.

- **VisiblePath**
  Property to set or determine a node parent's expand states.

- **WantTabs**
  Read or set whether the tree wants to process tabs on its own.

### Events

- **OnAdvancedHeaderDraw**
  Header paint support event.

- **OnAfterCellPaint**
  Paint support event.

- **OnAfterItemErase**
  Paint support event.

- **OnAfterItemPaint**
  Paint support event.
OnAfterPaint
Paint support event.

OnBeforeCellPaint
Paint support event.

OnBeforeItemErase
Paint support event.

OnBeforeItemPaint
Paint support event.

OnBeforePaint
Paint support event.

OnChange
Navigation support event.

OnChecked
Check support event.

OnChecking
Check support event.

OnCollapsed
Miscellaneous event.

OnCollapsing
Miscellaneous event.

OnColumnClick
Header and column support event.

OnColumnDbClick
Header and column support event.

OnColumnResize
Header and column support routine.

OnCompareNodes
Sort and search support event.

OnCreateDataObject
Drag'n drop support event.

OnCreateDragManager
Drag'n drop support event.

OnCreateEditor
Editing support event.

OnDragAllowed
Drag'n drop support event.

- **OnDragDrop**
  - Drag'n drop support event.
- **OnDragOver**
  - Drag'n drop support event.
- **OnDrawHint**
  - Triggered when a node hint or tooltip must be drawn.
- **OnDrawNode**
  - Triggered when a node must be drawn.
- **OnEdited**
  - Editing support event.
- **OnEditing**
  - Editing support event.
- **OnExpanded**
  - Miscellaneous event.
- **OnExpanding**
  - Miscellaneous event.
- **OnFocusChanged**
  - Navigation support event.
- **OnFocusChanging**
  - Navigation support event.
- **OnFreeNode**
  - Data management node.
- **OnGetCellIsEmpty**
  - Triggered when the tree control needs to know whether a given column is empty.
- **OnGetCursor**
  - Miscellaneous event.
- **OnGetHeaderCursor**
  - Header and column support event.
- **OnGetHelpContext**
  - Miscellaneous event.
- **OnGetHintSize**
  - Triggered when a node hint or tooltip is about to show.
- **OnGetImageIndex**
Display management event.

- **OnGetImageIndexEx**
  Not documented.

- **OnGetLineStyle**
  Display management event.

- **OnGetNodeDataSize**
  Data management event.

- **OnGetNodeWidth**
  Triggered when a node is about to be drawn.

- **OnGetPopupMenu**
  Miscellaneous event.

- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.

- **OnHeaderClick**
  Header & column support event.

- **OnHeaderDblClick**
  Header & column support event.

- **OnHeaderDragged**
  Header & column support event.

- **OnHeaderDraggedOut**
  Header & column support event.

- **OnHeaderDragging**
  Header & column support event.

- **OnHeaderDraw**
  Header & column support event.

- **OnHeaderDrawQueryElements**
  Header & column support event.

- **OnHeaderMouseDown**
  Header & column support event.

- **OnHeaderMouseMove**
  Header & column support event.

- **OnHeaderMouseUp**
  Header & column support event.

- **OnHotChange**
  Navigation support event.
OnIncrementalSearch
Miscellaneous event.

OnInitChildren
Node management event.

OnInitNode
Node management event.

OnKeyAction
Miscellaneous event.

OnLoadNode
Streaming support event.

OnMeasureItem
Miscellaneous event.

OnNodeCopied
Miscellaneous event.

OnNodeCopying
Miscellaneous event.

OnNodeMoved
Miscellaneous event.

OnNodeMoving
Miscellaneous event.

OnPaintBackground
Paint support event.

OnRenderOLEData
Drag'n drop and clipboard support event.

OnResetNode
Node management event.

OnSaveNode
Streaming support event.

OnScroll
Miscellaneous event.

OnShowScrollbar
Not documented.

OnStateChange
Miscellaneous event.

OnStructureChange
Miscellaneous event.

OnUpdating
Miscellaneous event.

**TCustomVirtualDrawTree Class**

- **OnDrawHint**
  Triggered when a node hint or tooltip must be drawn.
- **OnDrawNode**
  Triggered when a node must be drawn.
- **OnGetHintSize**
  Triggered when a node hint or tooltip is about to show.
- **OnGetNodeWidth**
  Triggered when a node is about to be drawn.

**TBaseVirtualTree Class**

- **OnAdvancedHeaderDraw**
  Header paint support event.
- **OnAfterCellPaint**
  Paint support event.
- **OnAfterItemErase**
  Paint support event.
- **OnAfterItemPaint**
  Paint support event.
- **OnAfterPaint**
  Paint support event.
- **OnBeforeCellPaint**
  Paint support event.
- **OnBeforeItemErase**
  Paint support event.
- **OnBeforeItemPaint**
  Paint support event.
- **OnBeforePaint**
  Paint support event.
- **OnChange**
Navigation support event.

- **OnChecked**
  Check support event.

- **OnChecking**
  Check support event.

- **OnCollapsed**
  Miscellaneous event.

- **OnCollapsing**
  Miscellaneous event.

- **OnColumnClick**
  Header and column support event.

- **OnColumnDbClick**
  Header and column support event.

- **OnColumnResize**
  Header and column support routine.

- **OnCompareNodes**
  Sort and search support event.

- **OnCreateDataObject**
  Drag'n drop support event.

- **OnCreateDragManager**
  Drag'n drop support event.

- **OnCreateEditor**
  Editing support event.

- **OnDragAllowed**
  Drag'n drop support event.

- **OnDragDrop**
  Drag'n drop support event.

- **OnDragOver**
  Drag'n drop support event.

- **OnEditCancelled**
  Editing support event.

- **OnEdited**
  Editing support event.

- **OnEditing**
  Editing support event.
OnExpanded
  Miscellaneous event.

OnExpanding
  Miscellaneous event.

OnFocusChanged
  Navigation support event.

OnFocusChanging
  Navigation support event.

OnFreeNode
  Data management node.

OnGetCellsIsEmpty
  Triggered when the tree control needs to know whether a given column is empty.

OnGetCursor
  Miscellaneous event.

OnGetHeaderCursor
  Header and column support event.

OnGetHelpContext
  Miscellaneous event.

OnGetImageIndex
  Display management event.

OnGetImageIndexEx
  Not documented.

OnGetLineStyle
  Display management event.

OnGetNodeDataSize
  Data management event.

OnGetPopupMenu
  Miscellaneous event.

OnGetUserClipboardFormats
  Drag'n drop and clipboard support event.

OnHeaderClick
  Header & column support event.

OnHeaderDblClick
  Header & column support event.
OnHeaderDragged
Header & column support event.

OnHeaderDraggedOut
Header & column support event.

OnHeaderDragging
Header & column support event.

OnHeaderDraw
Header & column support event.

OnHeaderDrawQueryElements
Header & column support event.

OnHeaderMouseDown
Header & column support event.

OnHeaderMouseMove
Header & column support event.

OnHeaderMouseUp
Header & column support event.

OnHotChange
Navigation support event.

OnIncrementalSearch
Miscellaneous event.

OnInitChildren
Node management event.

OnInitNode
Node management event.

OnKeyAction
Miscellaneous event.

OnLoadNode
Streaming support event.

OnMeasureItem
Miscellaneous event.

OnNodeCopied
Miscellaneous event.

OnNodeCopying
Miscellaneous event.

OnNodeMoved
Miscellaneous event.
Miscellaneous event.

OnNodeMoving
Miscellaneous event.

OnPaintBackground
Paint support event.

OnRenderOLEData
Drag'n drop and clipboard support event.

OnResetNode
Node management event.

OnSaveNode
Streaming support event.

OnScroll
Miscellaneous event.

OnShowScrollbar
Not documented.

OnStateChange
Miscellaneous event.

OnStructureChange
Miscellaneous event.

OnUpdating
Miscellaneous event.

Methods

GetOptionsClass
Customization helper to determine which options class the tree should use.

TCustomVirtualDrawTree Class

DoDrawHint
Overridable method which triggers OnDrawHint.

DoGetHintSize
Overridable method which triggers OnGetHintSize.

DoGetNodeWidth
Overridable method which triggers OnGetNodeWidth.
**DoPaintNode**
Overridable method which triggers OnPaintNode.

**TBaseVirtualTree Class**

- **AbsoluteIndex**
  Reads the overall index of a node.

- **AddChild**
  Creates and adds a new child node to given node.

- **AddFromStream**
  Adds the content from the given stream to the given node.

- **AddToSelection**
  Adds one or more nodes to the current selection.

- **AdjustPaintCellRect**
  Used in descendants to modify the clip rectangle of the current column while painting a certain node.

- **AdjustPanningCursor**
  Loads the proper cursor which indicates into which direction scrolling is done.

- **AdviseChangeEvent**
  Used to register a delayed change event.

- **AllocateInternalDataArea**
  Registration method to allocate tree internal data per node.

- **Animate**
  Support method for animated actions in the tree view.

- **Assign**
  Used to copy properties from another Virtual Treeview.

- **BeginDrag**
  Starts an OLE drag'n drop operation.

- **BeginSynch**
  Enters the tree into a special synchronized mode.

- **BeginUpdate**
  Locks the tree view to perform several update operations.

- **CalculateSelectionRect**
  Support method for draw selection.
- **CanAutoScroll**
  Determines whether the tree can currently auto scroll its window.

- **CancelCutOrCopy**
  Cancels any pending cut or copy clipboard operation.

- **CancelEditNode**
  Cancel the current edit operation, if there is any.

- **CanEdit**
  Determines whether a node can be edited or not.

- **CanFocus**
  Support method to determine whether the tree window can receive the input focus.

- **CanShowDragImage**
  Determines whether a drag image should be shown.

- **Change**
  Central method called when a node's selection state changes.

- **ChangeScale**
  Helper method called by the VCL when control resizing is due.

- **CheckParentCheckState**
  Helper method for recursive check state changes.

- **Clear**
  Clears the tree and removes all nodes.

- **ClearChecked**
  Not documented.

- **ClearSelection**
  Removes all nodes from the current selection.

- **ClearTempCache**
  Helper method to clear the internal temporary node cache.

- **ColumnIsEmpty**
  Used to determine if a cell is considered as being empty.

- **CopyTo**
  Copies **Source** and all its child nodes to **Target**.

- **CopyToClipBoard**
  Copies all currently selected nodes to the clipboard.

- **CountLevelDifference**
  Determines the level difference of two nodes.
CountVisibleChildren
Determines the number of visible child nodes of the given node.

Create
Constructor of the control

CreateParams
Prepares the creation of the controls window handle.

CreateWnd
Initializes data, which depends on the window handle.

CutToClipBoard
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties
Helper method to customize loading and saving persistent tree data.

DeleteChildren
Removes all child nodes from the given node.

DeleteNode
Removes the given node from the tree.

DeleteSelectedNodes
Removes all currently selected nodes form the tree.

Destroy
Destructor of the control.

DetermineHiddenChildrenFlag
Determines whether all children of a given node are hidden.

DetermineHiddenChildrenFlagAllNodes
Determines whether all children of all nodes are hidden.

DetermineHitPositionLTR
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineHitPositionRTL
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineNextCheckState
Not documented.

DetermineScrollDirections
Not documented.
DoAdvancedHeaderDraw
Not documented.

DoAfterCellPaint
Not documented.

DoAfterItemErase
Not documented.

DoAfterItemPaint
Not documented.

DoAfterPaint
Not documented.

DoAutoScroll
Enables or disables the auto scroll timer.

DoBeforeCellPaint
Not documented.

DoBeforeDrag
Not documented.

DoBeforeItemErase
Not documented.

DoBeforeItemPaint
Not documented.

DoBeforePaint
Not documented.

DoCancelEdit
Called when the tree should stop editing without accepting changed values.

DoCanEdit
Not documented.

DoChange
Not documented.

DoCheckClick
Not documented.

DoChecked
Not documented.

DoChecking
Not documented.
DoCollapsed
Not documented.
DoCollapsing
Not documented.
DoColumnClick
Not documented.
DoColumnDbIClick
Not documented.
DoColumnResize
Not documented.
DoCompare
Not documented.
DoCreateDataObject
Not documented.
DoCreateDragManager
Not documented.
DoCreateEditor
Not documented.
DoDragDrop
Not documented.
DoDragExpand
Not documented.
DoDragging
Internal method which handles drag' drop.
DoDragOver
Not documented.
DoEdit
Initiates editing of the currently set focused column and edit node.
DoEndDrag
Not documented.
DoEndEdit
Stops the current edit operation and takes over the new content.
DoExpanded
Not documented.
DoExpanding
Not documented.

DoFocusChange
Not documented.

DoFocusChanging
Not documented.

DoFocusNode
Internal method to set the focused node.

DoFreeNode
Not documented.

DoGetAnimationType
Determines the type of animation to be used.

DoGetCursor
Not documented.

DoGetHeaderCursor
Not documented.

DoGetImageIndex
Not documented.

DoGetLineStyle
Not documented.

DoGetNodeHint
Not documented.

DoGetNodeTooltip
Not documented.

DoGetNodeWidth
Overridable method which always returns 0.

DoGetPopupMenu
Overridable method which triggers the OnGetPopup event.

DoGetUserClipboardFormats
Not documented.

DoHeaderClick
Not documented.

DoHeaderDbClick
Not documented.

DoHeaderDragged
Not documented.
DoHeaderDraggedOut
Not documented.
DoHeaderDragging
Not documented.
DoHeaderDraw
Not documented.
DoHeaderDrawQueryElements
Not documented.
DoHeaderMouseDown
Not documented.
DoHeaderMouseMove
Not documented.
DoHeaderMouseUp
Not documented.
DoHotChange
Not documented.
DoIncrementalSearch
Not documented.
DoInitChildren
Not documented.
DoInitNode
Not documented.
DoKeyAction
Not documented.
DoLoadUserData
Not documented.
DoMeasureItem
Not documented.
DoNodeCopied
Not documented.
DoNodeCopying
Not documented.
DoNodeMoved
Not documented.
DoNodeMoving
Not documented.

DoPaintBackground
Not documented.

DoPaintDropMark
Overridable method which draws the small line on top of a nodes image depending on the current drop state.

DoPaintNode
Overridable method which does nothing.

DoPopupMenu
Overridable method which shows the popup menu for the given node.

DoRenderOLEData
Not documented.

DoReset
Not documented.

DoSaveUserData
Not documented.

DoScroll
Overridable method which triggers the OnScroll event.

DoSetOffsetXY
Internal core routine to set the tree's scroll position.

DoShowScrollbar
Not documented.

DoStartDrag
Not documented.

DoStateChange
Not documented.

DoStructureChange
Not documented.

DoTimerScroll
Callback method which is triggered whenever the scroll timer fires.

DoUpdating
Not documented.

DoValidateCache
Not documented.

DragCanceled
Called by the VCL when a drag'n drop operation was canceled by the user.

- **DragDrop**
  Helper method, which is used when a drag operation is finished.

- **DragEnter**
  Not documented.

- **DragFinished**
  Called when a drag operation is finished (accepted or cancelled).

- **Dragging**
  Returns true if a drag'n drop operation is in progress.

- **DragLeave**
  Not documented.

- **DragOver**
  Not documented.

- **DrawDottedHLine**
  Not documented.

- **DrawDottedVLine**
  Not documented.

- **EditNode**
  Starts editing the given node if allowed to.

- **EndEditNode**
  Stops node editing if it was started before.

- **EndSynch**
  Counterpart to **BeginSynch**.

- **EndUpdate**
  Resets the update lock set by **BeginUpdate**.

- **ExecuteAction**
  Not documented.

- **FindNodeInSelection**
  Helper method to find the given node in the current selection.

- **FinishChunkHeader**
  Not documented.

- **FinishCutOrCopy**
  Stops any pending cut or copy clipboard operation.

- **FlushClipboard**
Renders all pending clipboard data.

FontChanged
Not documented.

FullCollapse
Collapses all nodes in the tree.

FullExpand
Expands all nodes in the tree.

GetBorderDimensions
Not documented.

GetCheckImage
Not documented.

GetCheckImageListFor
Not documented.

GetColumnClass
Returns the class to be used to manage columns in the tree.

GetControlsAlignment
Not documented.

GetDisplayRect
Returns the visible region used by the given node in client coordinates.

GetFirst
Group of node navigation functions.

GetFirstChecked
Not documented.

GetFirstChild
Group of node navigation functions.

GetFirstCutCopy
Group of node navigation functions.

GetFirstInitialized
Group of node navigation functions.

GetFirstNoInit
Group of node navigation functions.

GetFirstSelected
Group of node navigation functions.

GetFirstVisible
Group of node navigation functions.

- **GetFirstChild**
  - Group of node navigation functions.
- **GetFirstChildNoInit**
  - Group of node navigation functions.
- **GetFirstChildNoInit**
  - Group of node navigation functions.

- **GetHeaderClass**
  - Returns the header class to be used by the tree.

- **GetHintWindowClass**
  - Not documented.

- **GetHitTestInfoAt**
  - Returns information about the node at the given position.

- **GetImageIndex**
  - Not documented.

- **GetLast**
  - Group of node navigation functions.

- **GetLastChild**
  - Group of node navigation functions.

- **GetLastChildNoInit**
  - Group of node navigation functions.

- **GetLastInitialized**
  - Group of node navigation functions.

- **GetLastNoInit**
  - Group of node navigation functions.

- **GetLastVisible**
  - Group of node navigation functions.

- **GetLastVisibleChild**
  - Group of node navigation functions.

- **GetLastVisibleChildNoInit**
  - Group of node navigation functions.

- **GetLastVisibleNoInit**
  - Group of node navigation functions.

- **GetMaxColumnWidth**
  - Returns the width of the largest node in the given column.
**GetMaxRightExtend**
Determines the maximum width of the currently visible part of the tree.

**GetNativeClipboardFormats**
Used to let descendants and the application add their own supported clipboard formats.

**GetNext**
Group of node navigation functions.

**GetNextChecked**
Not documented.

**GetNextCutCopy**
Group of node navigation functions.

**GetNextInitialized**
Group of node navigation functions.

**GetNextNoInit**
Group of node navigation functions.

**GetNextSelected**
Group of node navigation functions.

**GetNextSibling**
Group of node navigation functions.

**GetNextVisible**
Group of node navigation functions.

**GetNextVisibleNoInit**
Group of node navigation functions.

**GetNextVisibleSibling**
Group of node navigation functions.

**GetNextVisibleSiblingNoInit**
Group of node navigation functions.

**GetNodeAt**
Not documented.

**GetNodeData**
Returns the address of the user data area of the given node.

**GetNodeLevel**
Returns the indentation level of the given node.

**GetOptionsClass**
Customization helper to determine which options class the tree should
use.

- **GetPrevious**
  Group of node navigation functions.
- **GetPreviousInitialized**
  Group of node navigation functions.
- **GetPreviousNoInit**
  Group of node navigation functions.
- **GetPreviousSibling**
  Group of node navigation functions.
- **GetPreviousVisible**
  Group of node navigation functions.
- **GetPreviousVisibleNoInit**
  Group of node navigation functions.
- **GetPreviousVisibleSibling**
  Group of node navigation functions.
- **GetPreviousVisibleSiblingNoInit**
  Group of node navigation functions.
- **GetSortedCutCopySet**
  Returns a sorted list of nodes, which are marked for cut or copy clipboard operation.
- **GetSortedSelection**
  Returns a sorted list of all currently selected nodes.
- **GetTextInfo**
  Helper method for node editors, hints etc.
- **GetTreeFromDataObject**
  OLE drag'n drop and clipboard support method.
- **GetTreeRect**
  Returns the size of the virtual tree image.
- **GetVisibleParent**
  Returns the first (nearest) parent node, which is visible.
- **HandleHotTrack**
  Not documented.
- **HandleIncrementalSearch**
  Not documented.
- **HandleMouseDb1Click**
HandleMouseDown
Not documented.

HandleMouseUp
Not documented.

HasAsParent
Determines if the given node has got another node as one of its parents.

HasImage
Not documented.

HasPopupMenu
Determines whether there is a pop up menu assigned to the tree.

InitChildren
Not documented.

InitNode
Not documented.

InsertNode
Inserts a new node and returns it to the caller.

InternalAddFromStream
Not documented.

InternalAddToSelection
Not documented.

InternalCacheNode
Not documented.

InternalClearSelection
Not documented.

InternalConnectNode
Not documented.

InternalData
Returns the address of the internal data for a tree class.

InternalDisconnectNode
Not documented.

InternalRemoveFromSelection
Not documented.

InvalidateCache
Empties the internal node cache and marks it as invalid.

- **InvalidateChildren**
  Invalidates all children of the given node.

- **InvalidateColumn**
  Invalidates the client area part of a column.

- **InvalidateNode**
  Invalidates the given node.

- **InvalidateToBottom**
  Invalidates the client area starting with the top position of the given node.

- **InvertSelection**
  Inverts the current selection.

- **IsEditing**
  Tells the caller whether the tree is currently in edit mode.

- **IsMouseSelecting**
  Tell the caller whether the tree is currently in draw selection mode.

- **IterateSubtree**
  Iterator method to go through all nodes of a given sub tree.

- **Loaded**
  Not documented.

- **LoadFromFile**
  Loads previously streamed out tree data back in again.

- **LoadFromStream**
  Loads previously streamed out tree data back in again.

- **MainColumnInfoChanged**
  Not documented.

- **MarkCutCopyNodes**
  Not documented.

- **MeasureItemHeight**
  Not documented.

- **MouseMove**
  Not documented.

- **MoveTo**
  Moves **Source** and all its child nodes to **Target**.

- **Notification**
OriginalWMNCPaint
Not documented.

Paint
TControl's Paint method used here to display the tree.

PaintCheckImage
Not documented.

PaintImage
Not documented.

PaintNodeButton
Not documented.

PaintSelectionRectangle
Not documented.

PaintTree
Main paint routine for the tree image.

PaintTreeLines
Not documented.

PanningWindowProc
Not documented.

PasteFromClipboard
Inserts the content of the clipboard into the tree.

PrepareDragImage
Not documented.

Print
Not documented.

ProcessDrop
Helper method to ease OLE drag'n drop operations.

ProcessOLEData
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk
Not documented.

ReadNode
Not documented.

RedirectFontChangeEvent
Not documented.
- ReinitChildren
  Forces all child nodes of Node to be reinitialized.
- ReinitNode
  Forces a reinitialization of the given node.
- RemoveFromSelection
  Removes the given node from the current selection.
- RenderOLEData
  Renders pending OLE data.
- RepaintNode
  Causes the treeview to repaint the given node.
- ResetNode
  Resets the given node to uninitialized.
- ResetRangeAnchor
  Not documented.
- RestoreFontChangeEvent
  Not documented.
- SaveToFile
  Saves the entire content of the tree into a file or stream.
- SaveToStream
  Saves the entire content of the tree into a file or stream.
- ScrollIntoView
  Scrolls the tree so that the given node comes in the client area.
- SelectAll
  Selects all nodes in the tree.
- SelectNodes
  Selects a range of nodes.
- SetBiDiMode
  Not documented.
- SetFocusedNodeAndColumn
  Not documented.
- SkipNode
  Not documented.
- Sort
  Sorts the given node.
- SortTree
Sorts the entire tree view.

- **StartWheelPanning**
  Not documented.

- **StopWheelPanning**
  Not documented.

- **StructureChange**
  Not documented.

- **SuggestDropEffect**
  Not documented.

- **ToggleNode**
  Changes a node's expand state to the opposite state.

- **ToggleSelection**
  Toggles the selection state of a range of nodes.

- **UnselectNodes**
  Deselects a range of nodes.

- **UpdateAction**
  Not documented.

- **UpdateDesigner**
  Not documented.

- **UpdateEditBounds**
  Not documented.

- **UpdateHeaderRect**
  Not documented.

- **UpdateHorizontalScrollBar**
  Applies changes to the horizontal and vertical scrollbars.

- **UpdateScrollBars**
  Applies changes to the horizontal and vertical scrollbars.

- **UpdateVerticalScrollBar**
  Applies changes to the horizontal and vertical scrollbars.

- **UpdateWindowAndDragImage**
  Not documented.

- **UseRightToLeftReading**
  Helper method for right-to-left layout.

- **ValidateCache**
  Initiates the validation of the internal node cache.
**ValidateChildren**  
Validates all children of a given node.

**ValidateNode**  
Validates a given node.

**ValidateNodeDataSize**  
Helper method for node data size initialization.

**WndProc**  
Redirected window procedure to do some special processing.

**WriteChunks**  
Writes the core chunks for the given node to the given stream.

**WriteNode**  
Writes the cover (envelop) chunk for the given node to the given stream.

---

**Legend**

- **published**
- **Property**
- **public**
- **protected**
- **read only**
- **Event**
- **Method**
- **virtual**

**Class Hierarchy**

```
TCustomControl -> TBaseVirtualTree -> TCustomVirtualDrawTree -> TVirtualDrawTree
```

**File**  
VirtualTrees
**TVirtualDrawTree.Action Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property Action;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

`TVirtualDrawTree Class`

**Links**

`TVirtualDrawTree Class`

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.Align Property
TVirtualDrawTree Class

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

Links
TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.Alignment Property

TVirtualDrawTree Class

Determines the horizontal alignment of text if no columns are defined.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.Anchors Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property Anchors;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualDrawTree Class**

**Links**

**TVirtualDrawTree Class**

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.AnimationDuration Property

TVirtualDrawTree Class

Determines the maximum duration the tree can use to play an animation.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.AutoExpandDelay Property**

**TVirtualDrawTree Class**

Time delay after which a node gets expanded if it is the current drop target.

**Pascal**

```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*

**Links**

*TVirtualDrawTree Class*

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.AutoScrollDelay Property**

TVirtualDrawTree Class

Time which determines when auto scrolling should start.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.AutoScrollInterval Property**

**TVirtualDrawTree Class**

Time interval between scroll events when doing auto scroll.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.Background Property**

**TVirtualDrawTree Class**

Holds a background image for the tree.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.BackgroundOffsetX Property**

*TVirtualDrawTree Class*

Horizontal offset of the background image.

**Pascal**

```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*

**Links**

*TVirtualDrawTree Class*

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.BackgroundOffsetY Property

TVirtualDrawTree Class

Vertical offset of the background image.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.BevelEdges Property
TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property BevelEdges;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

*What do you think about this topic?* Send feedback!
**TVirtualDrawTree.BevelInner Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property BevelInner;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

---

_What do you think about this topic? _Send feedback!
TVirtualDrawTree.BevelKind Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property BevelKind;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.BevelOuter Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property BevelOuter;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.BevelWidth Property

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.BiDiMode Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property BiDiMode;
```

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TVirtualDrawTree Class

Links
TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.BorderStyle Property**

**TVirtualDrawTree Class**

Same as TForm.BorderStyle.

**Pascal**

```pascal
property BorderStyle: TBorderStyle;
```

**Description**

See TForm.BorderStyle.

**Class**

**TVirtualDrawTree Class**

**Links**

**TVirtualDrawTree Class**

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.BorderWidth Property**

**Class**

TVirtualDrawTree Class

Not documented.

**Pascal**

```pascal
property BorderRadius;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.ButtonFillMode Property**

**TVirtualDrawTree Class**

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

**Links**

TVirtualDrawTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.ButtonStyle** Property

**TVirtualDrawTree Class**

Determines the look of node buttons.

**Pascal**

```pascal
property ButtonStyle: TVTButtonStyle;
```

**Description**

Determines the look of node buttons.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

---

What do you think about this topic? Send feedback!
TVirtualDrawTree.Canvas Property

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.ChangeDelay Property**

*TVirtualDrawTree Class*

Time which determines when the OnChange event should be triggered after the actual change event.

**Pascal**

```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) happens. 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*

**Links**

*TVirtualDrawTree Class*
What do you think about this topic? Send feedback!
**TVirtualDrawTree.CheckImageKind Property**

TVirtualDrawTree Class

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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.ClipboardFormats Property**

**TVirtualDrawTree Class**

Special class to keep a list of clipboard format descriptions.

**Pascal**

```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

**Links**

TVirtualDrawTree Class
**TVirtualDrawTree.Color Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property Color;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualDrawTree Class**

**Links**

**TVirtualDrawTree Class**

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.Colors Property**

**TVirtualDrawTree Class**

A collection of colors used in the tree.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.Constraints Property**

**TVirtualDrawTree Class**

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

**Links**

TVirtualDrawTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.Ctl3D Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property Ctl3D;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.CustomCheckImages Property**

**TVirtualDrawTree Class | See Also**

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

**Links**

TVirtualDrawTree Class, See Also
**TVirtualDrawTree.DefaultNodeHeight Property**

*TVirtualDrawTree Class*

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*

**Links**

*TVirtualDrawTree Class*

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.DefaultPasteMode Property

TVirtualDrawTree Class | See Also

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

Links

TVirtualDrawTree Class, See Also
What do you think about this topic? Send feedback!
**TVirtualDrawTree.DragCursor Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property DragCursor;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.DragHeight Property

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.DragImageKind Property

TVirtualDrawTree Class

Read or set what should be shown in the drag image.

Pascal

```pascal
property DragImageKind: TVTDragImageKind;
```

Description

DragImageKind allows to switch parts of the drag image off and on.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.DragKind Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property DragKind;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.DragMode Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property DragMode;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.DragOperations Property
TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class
TVirtualDrawTree.DragType Property

Read or set which subsystem should be used for dragging.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.DragWidth Property

TVirtualDrawTree Class

Read or set the horizontal limit of the internal drag image.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.DrawSelectionMode**

**Property**

`TVirtualDrawTree Class`

Read or set how multiselection with the mouse is to be visualized.

**Pascal**

```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`

**Links**

`TVirtualDrawTree Class`

What do you think about this topic? Send feedback!
TVirtualDrawTree.EditDelay Property

Read or set the maximum time between two single clicks on the same node, which should start node editing.

**Pascal**

```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 maximum time distance between both clicks in which the edit operation is started.

**See Also**

Editors and editing

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class, See Also

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.Enabled Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property Enabled;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.Font Property

Same as TWinControl.Font.

Pascal

    property Font;

Description

See TWinControl.Font.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.Header Property

TVirtualDrawTree Class | See Also

Provides access to the header instance.

Pascal

```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

Links

TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualDrawTree.HintAnimation Property

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.HintMode Property

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.HotCursor Property**

**TVirtualDrawTree Class | See Also**

Read or set which cursor should be used for hot nodes.

**Pascal**

```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

**Links**

TVirtualDrawTree Class, See Also
**TVirtualDrawTree.Images Property**

**TVirtualDrawTree Class | See Also**

Read or set the tree's normal image list.

**Pascal**

```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

**Links**

TVirtualDrawTree Class, See Also

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.IncrementalSearch Property**

**TVirtualDrawTree Class | See Also**

Read or set the current incremental search mode.

**Pascal**

```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

**Links**

TVirtualDrawTree Class, See Also
**TVirtualDrawTree.IncrementalSearchDirection**

**Property**

Read or set the direction to be used for incremental search.

**Pascal**

```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

**Links**

TVirtualDrawTree Class, See Also

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.IncrementalSearchStart Property

TVirtualDrawTree Class | See Also

Read or set where to start incremental search.

Pascal

```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

Links

TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualDrawTree.IncrementalSearchTimeout Property

TVirtualDrawTree Class | See Also

Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Pascal

```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
TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualDrawTree.Indent Property

TVirtualDrawTree Class

Read or set the indentation amount for node levels.

Pascal

```pascal
property Indent: Cardinal;
```

Description

Each new level in the tree (child nodes of a parent node) are visually shifted to distinguish between them and their parent node (that's the tree layout after all). The Indent property determines the shift distance in pixels.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.LineMode Property
TVirtualDrawTree Class

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

Links
TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.LineStyle Property

TVirtualDrawTree Class

Read or set the mode of the tree lines.

Pascal

```pascal
propertyLineStyle: 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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.Margin Property**

TVirtualDrawTree Class | See Also

Read or set the tree's node margin.

**Pascal**

```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

**Links**
TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualDrawTree.NodeAlignment Property**

Read or set the node alignment value.

**Pascal**

```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

**Links**

TVirtualDrawTree Class, See Also
**TVirtualDrawTree.NodeDataSize Property**

**TVirtualDrawTree Class | See Also**

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

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class, See Also
**TVirtualDrawTree.OnAdvancedHeaderDraw Event**

**TVirtualDrawTree Class | See Also**

Header paint support event.

**Pascal**

```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

Links
TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnAfterCellPaint Event

TVirtualDrawTree Class | See Also

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

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
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

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class, See Also
TVirtualDrawTree.OnAfterItemPaint Event

TVirtualDrawTree Class | See Also

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

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class, See Also
TVirtualDrawTree.OnAfterPaint Event

Paint support event.

Pascal

```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

Class

- TVirtualDrawTree Class

Links

- TVirtualDrawTree Class, See Also
TVirtualDrawTree.OnBeforeCellPaint Event

TVirtualDrawTree Class | See Also

Paint support event.

Pascal

```
property OnBeforeCellPaint: TVTBeforeCellPaintEvent;
```

Description

This event is called immediately before a cell is painted.

See Also

Paint cycles and stages

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
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

Class
TVirtualDrawTree Class

Links
TVirtualDrawTree Class, See Also
**TVirtualDrawTree.OnBeforeItemPaint Event**

Pascal

```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

**Class**

- TVirtualDrawTree Class

**Links**

- TVirtualDrawTree Class, See Also

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnBeforePaint Event**

**TVirtualDrawTree Class | See Also**

Paint support event.

**Pascal**

```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

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class, See Also

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnChange Event**

**TVirtualDrawTree Class**

Navigation support event.

**Pascal**

```pascal
property OnChange: TVTChangeEvent;
```

**Description**

Called when a node's selection state has changed.

**Class**

**TVirtualDrawTree Class**

**Links**

**TVirtualDrawTree Class**

---

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnChecked Event**

**TVirtualDrawTree Class**

Check support event.

**Pascal**

```pascal
property OnChecked: TVTChangeEvent;
```

**Description**

Triggered when a node's check state has changed.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnChecking Event**

**TVirtualDrawTree Class**

Check support event.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnClick Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property OnClick;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnCollapsed Event**

**TVirtualDrawTree Class**

Miscellaneous event.

**Pascal**

```pascal
property OnCollapsed: TVTChangeEvent;
```

**Description**

Triggered after a node has been collapsed, that is, its child nodes are no longer displayed.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnCollapsing Event

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnColumnButtonClick Event**

**TVirtualDrawTree Class | See Also**

Header and column support event.

**Pascal**

```
property OnColumnButtonClick: TVTColumnButtonClickEvent;
```

**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**

OnHeaderButtonClick

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class, See Also

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnColumnDblClick Event

TVirtualDrawTree Class | See Also

Header and column support event.

Pascal

```
property OnColumnDblClick: TVTColumnDblClickEvent;
```

Description

Same as OnColumnClick but for double clicks.

See Also

OnColumnClick, OnHeaderDblClick

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnColumnResize Event**

TVirtualDrawTree Class

Header and column support routine.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnCompareNodes Event**

**TVirtualDrawTree Class | See Also**

Sort and search support event.

**Pascal**

```pascal
property OnCompareNodes: TVTCompareEvent;
```

**Description**

This event is the core event for all comparisons 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 are 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 comparison result as would there be an ascending sort order.

Below is some sample code taken from the Advanced Demo:
procedure TMainForm.VDT1CompareNodes(Sender: TBaseVirtualTree;
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

Links
TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnCreateDataObject Event

**TVirtualDrawTree Class**

Drag'n drop support event.

**Pascal**

```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. Descendants (which override DoGetDataObject) or the application can return an own IDataObject implementation to support special formats.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnCreateDragManager Event

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnCreateEditor Event

TVirtualDrawTree Class | See Also

Editing support event.

Pascal

```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. Descendants like TCustomVirtualStringTree supply a generic and simple string editor.

See Also

Editors and editing

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnDblClick Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property OnDblClick;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnDragAllowed Event**

Drag'n drop support event.

**Pascal**

```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 preferred over doing a **manual** BeginDrag.

**Notes**

The OnDragAllowed event is called only if the current DragMode is dmManual.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.onDragDrop Event

TVirtualDrawTree Class

Drag'n drop support event.

Pascal

```pascal
property OnDragDrop: TVTDragDropEvent;
```

Description

Triggered when either a VCL or a OLE drop action occurred. 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.

```pascal
procedure TMainForm.VTDragDrop(Sender: TBaseVirtualTree;
const Formats: array of Word;
Shift: TShiftState;
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 reject (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
            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;
  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
end;

Class

TVirtualDrawTree Class

Links
TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnDragOver Event

Drag'n drop support event.

Pascal

```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

Links

TVirtualDrawTree Class, See Also
**TVirtualDrawTree.OnDrawHint Event**

**TVirtualDrawTree Class**

Triggered when a node hint or tooltip must be drawn.

**Pascal**

```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**

**Links**

**TVirtualDrawTree Class**

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnDrawNode Event

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnEdited Event

Editing support event.

Pascal

```pascal
property OnEdited: TVTEditChangeEvent;
```

Description

Triggered when an edit action has successfully been finished.

See Also

Editors and editing

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class, See Also
TVirtualDrawTree.OnEditing Event

Editing support event.

Pascal

```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

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class, See Also
**TVirtualDrawTree.OnEndDock Property**

**TVirtualDrawTree Class**

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

**Links**

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnEndDrag Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property OnEndDrag;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

---

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnEnter Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property OnEnter;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualDrawTree Class**

**Links**

**TVirtualDrawTree Class**

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnExit Property**

*TVirtualDrawTree Class*

Not documented.

Pascal

```pascal
property OnExit;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

*TVirtualDrawTree Class*

**Links**

*TVirtualDrawTree Class*
TVirtualDrawTree.OnExpanded Event

TVirtualDrawTree Class

Missellaneous event.

Pascal

```pascal
property OnExpanded: TVTChangeEvent;
```

Description

Triggered after a node has been expanded.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnExpanding Event

TVirtualDrawTree Class

Miscellaneous event.

Pascal

```pascal
property OnExpanding: TVTChangingEvent;
```

Description

Triggered just before a node is expanded. Use **Allowed** to allow or deny this action.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

---

*What do you think about this topic?* Send feedback!
**TVirtualDrawTree.OnFocusChanged Event**

**TVirtualDrawTree Class**

Navigation support event.

**Pascal**

```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

**Links**

TVirtualDrawTree Class
**TVirtualDrawTree.OnFocusChanging Event**

**TVirtualDrawTree Class**

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`

**Links**

`TVirtualDrawTree Class`

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnFreeNode Event**

*TVirtualDrawTree Class*

- Data management node.

**Pascal**

```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`

**Links**

`TVirtualDrawTree Class`

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnGetCellIsEmpty Event**

**TVirtualDrawTree Class**

Triggered when the tree control needs to know whether a given column is empty.

**Pascal**

```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

**Links**

TVirtualDrawTree Class
What do you think about this topic? Send feedback!
TVirtualDrawTree.OnGetCursor Event

TVirtualDrawTree Class

Miscellaneous event.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnGetHeaderCursor**

**Event**

*TVirtualDrawTree Class*

Header and column support event.

**Pascal**

```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*

**Links**

*TVirtualDrawTree Class*

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnGetHelpContext Event

TVirtualDrawTree Class

Miscellaneous event.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnGetHintSize Event**

**TVirtualDrawTree Class**

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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnGetImageIndex Event

TVirtualDrawTree Class

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 they 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

Links
TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnGetImageIndexEx Event

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnGetLineStyle Event

TVirtualDrawTree Class | See Also

Display management event.

Pascal

```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 IsCustomStyle. 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

Links
  TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualDrawTree.**

### OnGetNodeDataSize Event

**TVirtualDrawTree Class | See Also**

Data management event.

**Pascal**

```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**

**Class**

**TVirtualDrawTree Class**
Links
TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnGetNodeWidth Event**

**TVirtualDrawTree Class**

Triggered when a node is about to be drawn.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnGetPopupMenu Event

TVirtualDrawTree Class

Miscellaneous event.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnGetUserClipboardFormats**

**Event**

*TVirtualDrawTree Class*

Drag'n drop and clipboard support event.

**Pascal**

```pascal
property OnGetUserClipboardFormats: TVTGetUserClipboardFormats;
```

**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*
**TVirtualDrawTree.OnHeaderClick Event**

**TVirtualDrawTree Class | See Also**

Header & column support event.

**Pascal**

```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

**Links**

TVirtualDrawTree Class, See Also

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnHeaderDblClick Event**

**TVirtualDrawTree Class | See Also**

Header & column support event.

**Pascal**

```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

**Links**

TVirtualDrawTree Class, See Also

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnHeaderDragged Event**

**TVirtualDrawTree Class**

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**

**Links**

**TVirtualDrawTree Class**

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnHeaderDraggedOut Event**

**TVirtualDrawTree Class**

Header & column support event.

**Pascal**

```pascal
property OnHeaderDraggedOut: TTVTHeaderDraggedOutEvent;
```

**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

**Links**

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnInitHeaderDragging Event**

**TVirtualDrawTree Class**

Header & column support event.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnHeaderDraw Event**

*TVirtualDrawTree Class*

Header & column support event.

**Pascal**

```pascal
property OnHeaderDraw: TVTHeaderPaintEvent;
```

**Description**

If you set the hoOwnerDraw style in TVTHHeader.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*

**Links**

*TVirtualDrawTree Class*

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnHeaderDrawQueryElements Event

TVirtualDrawTree Class | See Also

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

Links

TVirtualDrawTree Class, See Also
TVirtualDrawTree.OnHeaderMouseDown Event

TVirtualDrawTree Class

Header & column support event.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnHeaderMouseMove Event

TVirtualDrawTree Class

Header & column support event.

Pascal

```pascal
property OnHeaderMouseMove: TVTHeaderMouseMoveEvent;
```

Description

This event is triggered when the mouse pointer is moved over the header area.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnHeaderMouseMove Event

TVirtualDrawTree Class

Header & column support event.

Pascal

```
property OnHeaderMouseMove: TVTHeaderMouseEvent;
```

Description

This event is very much like OnHeaderMouseDown but is triggered when a mouse button is released.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class
**TVirtualDrawTree.OnHotChange Event**

**TVirtualDrawTree Class**

Navigation support event.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnIncrementalSearch Event
TVirtualDrawTree Class

Miscellaneous event.

Pascal

```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: TNotifyEvent; var Result: Integer);

var
  S, PropText: string;

begin
  // Note: This code requires a proper Unicode/Wide
```
// size and clarity reasons. For now strings are
// 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 StrLCmp we can specify a maximum length
// which match only partially.
Result := StrLCmp(PChar(S), PChar(PropText), Min(Length(S), Length(PropText)));

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

Links
TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnInitChildren Event**

**TVirtualDrawTree Class | See Also**

Node management event.

**Pascal**

```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 initialization without 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

**Class**

- TVirtualDrawTree Class

**Links**

- TVirtualDrawTree Class, See Also
What do you think about this topic? Send feedback!
TVirtualDrawTree.OnInitNode Event

Node management event.

Pascal

```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

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnKeyAction Event**

**TVirtualDrawTree Class**

Miscellaneous event.

**Pascal**

```
property OnKeyAction: TVTKeyActionEvent;
```

**Description**

This event is a convenient way for the application or descendant 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

**Links**

TVirtualDrawTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnKeyDown Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property OnKeyDown;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualDrawTree Class**

**Links**

**TVirtualDrawTree Class**

*What do you think about this topic?* Send feedback!
TVirtualDrawTree.OnKeyPress Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property OnKeyPress;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnKeyUp Property**

**TVirtualDrawTree Class**

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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnLoadNode** Event

*TVirtualDrawTree Class | See Also*

Streaming support event.

**Pascal**

```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*

**Links**

*TVirtualDrawTree Class, See Also*
What do you think about this topic? Send feedback!
TVirtualDrawTree.OnMeasureItem Event

TVirtualDrawTree Class | See Also

Miscellaneous event.

Pascal

```pascal
property OnMeasureItem: TVTMeasureItemEvent;
```

Description

Virtual Treeview supports individual node heights. However it might sometimes impractical 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
Links
TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnMouseDown

Property
TVirtualDrawTree Class

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

Links
TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnMouseMove Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property OnMouseMove;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnMouseUp Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property OnMouseUp;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnMouseWheel Property

Not documented.

Pascal

```pascal
property OnMouseWheel;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnNodeCopied Event**

**TVirtualDrawTree Class**

Miscellaneous event.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnNodeCopying Event**

**TVirtualDrawTree Class**

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

**Links**

TVirtualDrawTree Class

---

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnNodeMoved Event**

**TVirtualDrawTree Class**

Miscellaneous event.

**Pascal**

```pascal
property OnNodeMoved: TVTNodeMovedEvent;
```

**Description**

This event is very much like OnNodeCopied but used for moving nodes instead.

**Class**

**TVirtualDrawTree Class**

**Links**

**TVirtualDrawTree Class**

What do you think about this topic? Send feedback!
```
property OnNodeMoving: TVTNodeMovingEvent;
```

**Description**
This event is very much like OnNodeCopying but used for moving nodes instead.

**Class**
TVirtualDrawTree Class

**Links**
TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnPaintBackground Event

TVirtualDrawTree Class

Paint support event.

Pascal

```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 customization. For the free area in the tree window there is this event.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnRenderOLEData

**Event**

**TVirtualDrawTree Class**

Drag'n drop and clipboard support event.

**Pascal**

```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 descendents) have specified their own formats in `OnGetUserClipboardFormats`.

**Class**

**TVirtualDrawTree Class**

**Links**

**TVirtualDrawTree Class**

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnResetNode Event

Node management event.

Pascal

```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

Links

TVirtualDrawTree Class, See Also
TVirtualDrawTree.OnResize Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property OnResize;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnSaveNode Event

TVirtualDrawTree Class | See Also

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

**Links**
TVirtualDrawTree Class, See Also

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.OnScroll Event

TVirtualDrawTree Class | See Also

Miscellaneous event.

Pascal

```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

Links

TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnShowScrollbar Event

Not documented.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.OnStartDock Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property OnStartDock;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.OnStateChange Event**

TVirtualDrawTree Class

**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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.OnStructureChange Event**

**TVirtualDrawTree Class**

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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
TVirtualDrawTree. OnUpdating Event

TVirtualDrawTree Class

Miscellaneous event.

Pascal

```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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.ParentBiDiMode Property**

Class: TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property ParentBiDiMode;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class


What do you think about this topic? Send feedback!
**TVirtualDrawTree.ParentColor Property**

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property ParentColor;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree. ParentCtl3D Property

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.ParenFont Property

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.ParentShowHint Property**

**TVirtualDrawTree Class**

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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.PopupMenu Property

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**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

**Links**

TVirtualDrawTree Class

*What do you think about this topic? Send feedback!*
**TVirtualDrawTree.ScrollBarOptions Property**

**TVirtualDrawTree Class**

Reference to the scroll bar options class.

**Pascal**

```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

**Links**

TVirtualDrawTree Class

---

*What do you think about this topic?*  Send feedback!
**TVirtualDrawTree.SelectionBlendFactor Property**

TVirtualDrawTree Class | See Also

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

Links
TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualDrawTree.SelectionCurveRadius Property

TVirtualDrawTree Class | See Also

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

Links

TVirtualDrawTree Class, See Also
**TVirtualDrawTree.ShowHint Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property ShowHint;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

---

*What do you think about this topic? Send feedback!*
TVirtualDrawTree.StateImages Property

TVirtualDrawTree Class | See Also

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

Links

TVirtualDrawTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualDrawTree.TabOrder Property

TVirtualDrawTree Class

Not documented.

Pascal

```pascal
property TabOrder;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.TabStop Property

TVirtualDrawTree Class

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

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualDrawTree.TextMargin Property

TVirtualDrawTree Class | See Also

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

Links

TVirtualDrawTree Class, See Also
What do you think about this topic? Send feedback!
TVirtualDrawTree.TreeOptions Property

reference to the tree's options.

Pascal

```pascal
property TreeOptions: TVirtualTreeOptions;
```

Description

The tree options are one of the main switches to modify a treeview's behavior. Virtual Treeview supports customizing tree options by descendents. 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 descendents.

Class

TVirtualDrawTree Class

Links

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.Visible Property**

**TVirtualDrawTree Class**

Not documented.

**Pascal**

```pascal
property Visible;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualDrawTree Class

**Links**

TVirtualDrawTree Class

What do you think about this topic? Send feedback!
**TVirtualDrawTree.WantTabs Property**

**TVirtualDrawTree Class**

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

**Links**

TVirtualDrawTree Class

---

What do you think about this topic? Send feedback!
**TVirtualDrawTree.GetOptionsClass**

**Method**

*TVirtualDrawTree Class*

Customization helper to determine which options class the tree should use.

**Pascal**

```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 GetOptionsClass write SetOptionsClass;
  ...
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
  ...
end;
Class
TVirtualDrawTree Class

Links
TVirtualDrawTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree Class

Descentant of TBaseVirtualTree which is able to manage node captions on its own.

Pascal

```pascal
TVirtualStringTree = class(TCustomVirtualStringTree)
```

Description

TVirtualStringTree adds no new functionality to TCustomVirtualStringTree but is publicly available version and appears in the component palette.

Group

Classes

Members

Properties

- Action
  - Not documented.
- Align
  - Not documented.
- Alignment
  - Determines the horizontal alignment of text if no columns are defined.
- Anchors
  - Not documented.
- AnimationDuration
  - Determines the maximum duration the tree can use to play an
animation.

AutoExpandDelay
Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay
Time which determines when auto scrolling should start.

AutoScrollInterval
Time interval between scroll events when doing auto scroll.

Background
Holds a background image for the tree.

BackgroundOffsetX
Horizontal offset of the background image.

BackgroundOffsetY
Vertical offset of the background image.

BevelEdges
Not documented.

BevelInner
Not documented.

BevelKind
Not documented.

BevelOuter
Not documented.

BevelWidth
Not documented.

BiDiMode
Not documented.

BorderStyle
Same as TForm.BorderStyle.

BorderWidth
Not documented.

ButtonFillMode
Determines how to fill the background of the node buttons.

ButtonStyle
Determines the look of node buttons.

Canvas
Not documented.

- **ChangeDelay**
  Time which determines when the OnChange event should be triggered after the actual change event.

- **CheckImageKind**
  Determines which images should be used for checkboxes and radio buttons.

- **ClipboardFormats**
  Special class to keep a list of clipboard format descriptions.

- **Color**
  Not documented.

- **Colors**
  A collection of colors used in the tree.

- **Constraints**
  Not documented.

- **Ctl3D**
  Not documented.

- **CustomCheckImages**
  Assign your own image list to get the check images you like most.

- **DefaultNodeHeight**
  Read or set the height new nodes get as initial value.

- **DefaultPasteMode**
  Read or set the value, which determines where to add pasted nodes to.

- **DefaultText**
  Not documented.

- **DragCursor**
  Not documented.

- **DragHeight**
  Read or set the vertical limit of the internal drag image.

- **DragImageKind**
  Read or set what should be shown in the drag image.

- **DragKind**
  Not documented.

- **DragMode**
Not documented.

- **DragOperations**
  Read or set which drag operations may be allowed in the tree.

- **DragType**
  Read or set which subsystem should be used for dragging.

- **DragWidth**
  Read or set the horizontal limit of the internal drag image.

- **DrawSelectionMode**
  Read or set how multiselection with the mouse is to be visualized.

- **EditDelay**
  Read or set the maximum time between two single clicks on the same node, which should start node editing.

- **Enabled**
  Not documented.

- **Font**
  Same as TWinControl.Font.

- **Header**
  Provides access to the header instance.

- **HintAnimation**
  Read or set the current hint animation type.

- **HintMode**
  Read or set what type of hint you want for the tree view.

- **HotCursor**
  Read or set which cursor should be used for hot nodes.

- **Images**
  Read or set the tree's normal image list.

- **IncrementalSearch**
  Read or set the current incremental search mode.

- **IncrementalSearchDirection**
  Read or set the direction to be used for incremental search.

- **IncrementalSearchStart**
  Read or set where to start incremental search.

- **IncrementalSearchTimeout**
  Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.
Indent
Read or set the indentation amount for node levels.

LineMode
Read or set the mode of the tree lines.

LineStyle
Read or set the mode of the tree lines.

Margin
Read or set the tree's node margin.

NodeAlignment
Read or set the node alignment value.

NodeDataSize
Read or set the extra data size for each node.

OnClick
Not documented.

OnDblClick
Not documented.

OnEndDock
Not documented.

OnEndDrag
Not documented.

OnEnter
Not documented.

OnExit
Not documented.

OnKeyDown
Not documented.

OnKeyPress
Not documented.

OnKeyUp
Not documented.

OnMouseDown
Not documented.

OnMouseMove
Not documented.

OnMouseUp
Not documented.

**OnMouseWheel**
Not documented.

**OnResize**
Not documented.

**OnStartDock**
Not documented.

**OnStartDrag**
Not documented.

**ParentBiDiMode**
Not documented.

**ParentColor**
Not documented.

**ParentCtl3D**
Not documented.

**ParentFont**
Not documented.

**ParentShowHint**
Not documented.

**PopupMenu**
Not documented.

**RootNodeCount**
Read or set the number of nodes on the top level.

**ScrollBarOptions**
Reference to the scroll bar options class.

**SelectionBlendFactor**
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

**SelectionCurveRadius**
Read or set the current corner radius for node selection rectangles.

**ShowHint**
Not documented.

**StateImages**
Reference to the images list which is used for the state images.

**TabOrder**
Not documented.

- **TabStop**
  Not documented.
- **TextMargin**
  Read or set the distance of the node caption to its borders.
- **TreeOptions**
  Reference to the tree's options.
- **Visible**
  Not documented.
- **WantTabs**
  Read or set whether the tree wants to process tabs on its own.

### TCustomVirtualStringTree Class

- **DefaultText**
  Not documented.
- **EllipsisWidth**
  Not documented.
- **Text**
  Not documented.
- **TreeOptions**
  Reference to the tree's options.

### TBaseVirtualTree Class

- **Alignment**
  Determines the horizontal alignment of text if no columns are defined.
- **AnimationDuration**
  Determines the maximum duration the tree can use to play an animation.
- **AutoExpandDelay**
  Time delay after which a node gets expanded if it is the current drop target.
- **AutoScrollDelay**
  Time which determines when auto scrolling should start.
- **AutoScrollInterval**
Time interval between scroll events when doing auto scroll.

- **Background**
  Holds a background image for the tree.

- **BackgroundOffsetX**
  Horizontal offset of the background image.

- **BackgroundOffsetY**
  Vertical offset of the background image.

- **BorderStyle**
  Same as TForm.BorderStyle.

- **ButtonFillMode**
  Determines how to fill the background of the node buttons.

- **ButtonStyle**
  Determines the look of node buttons.

- **ChangeDelay**
  Time which determines when the **OnChange** event should be triggered after the actual change event.

- **CheckImageKind**
  Determines which images should be used for checkboxes and radio buttons.

- **CheckImages**
  Not documented.

- **CheckState**
  Read or set the check state of a node.

- **CheckType**
  Read or set the check type of a node.

- **ChildCount**
  Read or set the number of child nodes of a node.

- **ChildrenInitialized**
  Read whether a node’s child count has been initialized already.

- **ClipboardFormats**
  Special class to keep a list of clipboard format descriptions.

- **Colors**
  A collection of colors used in the tree.

- **CustomCheckImages**
  Assign your own image list to get the check images you like most.
DefaultNodeHeight
Read or set the height new nodes get as initial value.

DefaultPasteMode
Read or set the value, which determines where to add pasted nodes to.

DragHeight
Read or set the vertical limit of the internal drag image.

DragImage
Holds the instance of the internal drag image.

DragImageKind
Read or set what should be shown in the drag image.

DragManager
Holds the reference to the internal drag manager.

DragOperations
Read or set which drag operations may be allowed in the tree.

DragSelection
Keeps a temporary list of nodes during drag'n drop.

DragType
Read or set which subsystem should be used for dragging.

DragWidth
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode
Read or set how multiselection with the mouse is to be visualized.

DropTargetNode
Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

EditColumn
Not documented.

EditDelay
Read or set the maximum time between two single clicks on the same node, which should start node editing.

EditLink
Keeps a reference to the internal edit link during a node edit operation.

Expanded
Read or set the expanded state of a particular node.

- **FocusedColumn**
  Read or set the currently focused column.

- **FocusedNode**
  Read or set the currently focused node.

- **Font**
  Same as TWinControl.Font.

- **FullyVisible**
  Read or set whether a node is fully visible or not.

- **HasChildren**
  Read or set whether a node has got children.

- **Header**
  Provides access to the header instance.

- **HeaderRect**
  Returns the non-client-area rectangle used for the header.

- **HintAnimation**
  Read or set the current hint animation type.

- **HintMode**
  Read or set what type of hint you want for the tree view.

- **HotCursor**
  Read or set which cursor should be used for hot nodes.

- **HotNode**
  Read, which node is currently the hot node.

- **Images**
  Read or set the tree's normal image list.

- **IncrementalSearch**
  Read or set the current incremental search mode.

- **IncrementalSearchDirection**
  Read or set the direction to be used for incremental search.

- **IncrementalSearchStart**
  Read or set where to start incremental search.

- **IncrementalSearchTimeout**
  Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

- **Indent**
Read or set the indentation amount for node levels.

- **IsDisabled**
  Read or set the enabled state of the given node.

- **IsVisible**
  Read or set the visibility state of the given node.

- **LastClickPos**
  Used for retained drag start and wheel mouse scrolling.

- **LastDropMode**
  Read how the last drop operation finished.

- **LineMode**
  Read or set the mode of the tree lines.

- **LineStyle**
  Read or set the mode of the tree lines.

- **Margin**
  Read or set the tree's node margin.

- **MultiLine**
  Read or toggle the multiline feature for a given node.

- **NodeAlignment**
  Read or set the node alignment value.

- **NodeDataSize**
  Read or set the extra data size for each node.

- **NodeHeight**
  Read or set a node's height.

- **NodeParent**
  Read or set a node's parent node.

- **OffsetX**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetXY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **RootNode**
  Reference to the internal root node which is the anchor of the entire tree node hierarchy.

- **RootNodeCount**
Read or set the number of nodes on the top level.

- **ScrollBarOptions**
  Reference to the scroll bar options class.

- **SearchBuffer**
  Current input string for incremental search.

- **Selected**
  Property to modify or determine the selection state of a node.

- **SelectedCount**
  Contains the number of selected nodes.

- **SelectionBlendFactor**
  Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

- **SelectionCurveRadius**
  Read or set the current corner radius for node selection rectangles.

- **StateImages**
  Reference to the images list which is used for the state images.

- **TextMargin**
  Read or set the distance of the node caption to its borders.

- **TopNode**
  The top node is the node which is currently at the top border of the client area.

- **TotalCount**
  Returns the number of nodes in the tree.

- **TotalInternalDataSize**
  Keeps the currently accumulated data size for one node.

- **TreeOptions**
  Reference to the tree's options.

- **TreeStates**
  Property which keeps a set of flags which indicate current operation and states of the tree.

- **UpdateCount**
  Not documented.

- **VerticalAlignment**
  Used to set a node's vertical button alignment with regard to the entire node rectangle.
VisibleCount
Number of currently visible nodes.

VisiblePath
Property to set or determine a node parent's expand states.

WantTabs
Read or set whether the tree wants to process tabs on its own.

Events

OnAdvancedHeaderDraw
Header paint support event.

OnAfterCellPaint
Paint support event.

OnAfterItemErase
Paint support event.

OnAfterItemPaint
Paint support event.

OnAfterPaint
Paint support event.

OnBeforeCellPaint
Paint support event.

OnBeforeItemErase
Paint support event.

OnBeforeItemPaint
Paint support event.

OnBeforePaint
Paint support event.

OnChange
Navigation support event.

OnChecked
Check support event.

OnChecking
Check support event.

OnCollapsed
Miscellaneous event.
- **OnCollapsing**
  Miscellaneous event.
- **OnColumnClick**
  Header and column support event.
- **OnColumnDbClick**
  Header and column support event.
- **OnColumnResize**
  Header and column support routine.
- **OnCompareNodes**
  Sort and search support event.
- **OnCreateDataObject**
  Drag'n drop support event.
- **OnCreateDragManager**
  Drag'n drop support event.
- **OnCreateEditor**
  Editing support event.
- **OnDragAllowed**
  Drag'n drop support event.
- **OnDragDrop**
  Drag'n drop support event.
- **OnDragOver**
  Drag'n drop support event.
- **OnEditCancelled**
  Editing support event.
- **OnEdited**
  Editing support event.
- **OnEditing**
  Editing support event.
- **OnExpanded**
  Miscellaneous event.
- **OnExpanding**
  Miscellaneous event.
- **OnFocusChanged**
  Navigation support event.
- **OnFocusChanging**
Navigation support event.

- **OnFreeNode**
  Data management node.

- **OnGetCellsIsEmpty**
  Triggered when the tree control needs to know whether a given column is empty.

- **OnGetCursor**
  Miscellaneous event.

- **OnGetHeaderCursor**
  Header and column support event.

- **OnGetHelpContext**
  Miscellaneous event.

- **OnGetHint**
  Virtual string tree event to query for a custom hint text.

- **OnGetImageIndex**
  Display management event.

- **OnGetImageIndexEx**
  Not documented.

- **OnGetLineStyle**
  Display management event.

- **OnGetNodeDataSize**
  Data management event.

- **OnGetPopupMenu**
  Miscellaneous event.

- **OnGetText**
  Virtual string tree event to query for a node's normal or static text.

- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.

- **OnHeaderClick**
  Header & column support event.

- **OnHeaderDoubleClick**
  Header & column support event.

- **OnHeaderDragged**
  Header & column support event.

- **OnHeaderDraggedOut**
Header & column support event.

OnHeaderDragging
Header & column support event.

OnHeaderDraw
Header & column support event.

OnHeaderDrawQueryElements
Header & column support event.

OnHeaderMouseDown
Header & column support event.

OnHeaderMouseMove
Header & column support event.

OnHeaderMouseUp
Header & column support event.

OnHotChange
Navigation support event.

OnIncrementalSearch
Miscellaneous event.

OnInitChildren
Node management event.

OnInitNode
Node management event.

OnKeyAction
Miscellaneous event.

OnLoadNode
Streaming support event.

OnMeasureItem
Miscellaneous event.

OnNewText
Virtual string tree event to pass edited text.

OnNodeCopied
Miscellaneous event.

OnNodeCopying
Miscellaneous event.

OnNodeMoved
Miscellaneous event.
OnNodeMoving
Miscellaneous event.
OnPaintBackground
Paint support event.
OnPaintText
Event to change text formatting for particular nodes.
OnRenderOLEData
Drag’n drop and clipboard support event.
OnResetNode
Node management event.
OnSaveNode
Streaming support event.
OnScroll
Miscellaneous event.
OnShortenString
String tree event for custom handling of string abbreviations.
OnShowScrollbar
Not documented.
OnStateChange
Miscellaneous event.
OnStructureChange
Miscellaneous event.
OnUpdating
Miscellaneous event.

TCustomVirtualStringTree Class

OnGetHint
Virtual string tree event to query for a custom hint text.
OnGetText
Virtual string tree event to query for a node’s normal or static text.
OnNewText
Virtual string tree event to pass edited text.
OnPaintText
Event to change text formatting for particular nodes.
OnShortenString
String tree event for custom handling of string abbreviations.

TBaseVirtualTree Class

OnAdvancedHeaderDraw
Header paint support event.

OnAfterCellPaint
Paint support event.

OnAfterItemErase
Paint support event.

OnAfterItemPaint
Paint support event.

OnAfterPaint
Paint support event.

OnBeforeCellPaint
Paint support event.

OnBeforeItemErase
Paint support event.

OnBeforeItemPaint
Paint support event.

OnBeforePaint
Paint support event.

OnChange
Navigation support event.

OnChecked
Check support event.

OnChecking
Check support event.

OnCollapsed
Miscellaneous event.

OnCollapsing
Miscellaneous event.

OnColumnClick
Header and column support event.
OnColumnDb1Click
Header and column support event.

OnColumnResize
Header and column support routine.

OnCompareNodes
Sort and search support event.

OnCreateDataObject
Drag'n drop support event.

OnCreateDragManager
Drag'n drop support event.

OnCreateEditor
Editing support event.

OnDragAllowed
Drag'n drop support event.

OnDragDrop
Drag'n drop support event.

OnDragOver
Drag'n drop support event.

OnEditCancelled
Editing support event.

OnEdited
Editing support event.

OnEditing
Editing support event.

OnExpanded
Miscellaneous event.

OnExpanding
Miscellaneous event.

OnFocusChanged
Navigation support event.

OnFocusChanging
Navigation support event.

OnFreeNode
Data management node.

OnGetCellIsEmpty
Triggered when the tree control needs to know whether a given column is empty.

- **OnGetCursor**: Miscellaneous event.
- **OnGetHeaderCursor**: Header and column support event.
- **OnGetHelpContext**: Miscellaneous event.
- **OnGetImageIndex**: Display management event.
- **OnGetImageIndexEx**: Not documented.
- **OnGetLineStyle**: Display management event.
- **OnGetNodeDataSize**: Data management event.
- **OnGetPopupMenu**: Miscellaneous event.
- **OnGetUserClipboardFormats**: Drag'n drop and clipboard support event.
- **OnHeaderClick**: Header & column support event.
- **OnHeaderDbClick**: Header & column support event.
- **OnHeaderDragged**: Header & column support event.
- **OnHeaderDraggedOut**: Header & column support event.
- **OnHeaderDragging**: Header & column support event.
- **OnHeaderDraw**: Header & column support event.
- **OnHeaderDrawQueryElements**: Header & column support event.
- **OnHeaderMouseDown**
Header & column support event.

- **OnHeaderMouseMove**
- **OnHeaderMouseUp**

Header & column support event.

- **OnHotChange**

Navigation support event.

- **OnIncrementalSearch**

Miscellaneous event.

- **OnInitChildren**

Node management event.

- **OnInitNode**

Node management event.

- **OnKeyAction**

Miscellaneous event.

- **OnLoadNode**

Streaming support event.

- **OnMeasureItem**

Miscellaneous event.

- **OnNodeCopied**

Miscellaneous event.

- **OnNodeCopying**

Miscellaneous event.

- **OnNodeMoved**

Miscellaneous event.

- **OnNodeMoving**

Miscellaneous event.

- **OnPaintBackground**

Paint support event.

- **OnRenderOLEData**

Drag'n drop and clipboard support event.

- **OnResetNode**

Node management event.

- **OnSaveNode**

Streaming support event.
OnScroll
Miscellaneous event.

OnShowScrollbar
Not documented.

OnStateChange
Miscellaneous event.

OnStructureChange
Miscellaneous event.

OnUpdating
Miscellaneous event.

Methods

GetOptionsClass
Customization helper to determine which options class the tree should use.

TCustomVirtualStringTree Class

AdjustPaintCellRect
Method which can be used by descendents to adjust the given rectangle during a paint cycle.

CalculateTextWidth
Not documented.

ColumnIsEmpty
Used to determine if a cell is considered as being empty.

ComputeNodeHeight
Not documented.

ContentToClipboard
Not documented.

ContentToHTML
Not documented.

ContentToRTF
Not documented.

ContentToText
Not documented.
ContentToUnicode
Not documented.

Create
Constructor of the control

DefineProperties
Helper method to customize loading and saving persistent tree data.

DoCreateEditor
Not documented.

DoGetNodeHint
Not documented.

DoGetNodeTooltip
Not documented.

DoGetNodeWidth
Overridable method which always returns 0.

DoGetText
Not documented.

DoIncrementalSearch
Not documented.

DoNewText
Not documented.

DoPaintNode
Overridable method which does nothing.

DoPaintText
Not documented.

DoShortenString
Not documented.

DoTextDrawing
Not documented.

DoTextMeasuring
Not documented.

GetOptionsClass
Customization helper to determine which options class the tree should use.

GetTextInfo
Helper method for node editors, hints etc.
InternalData
Returns the address of the internal data for a tree class.

InvalidateNode
Invalidates the given node.

MainColumnChanged
Not documented.

Path
Not documented.

ReadChunk
Not documented.

ReadOldStringOptions
Not documented.

ReinitNode
Forces a reinitialization of the given node.

RenderOLEData
Renders pending OLE data.

WriteChunks
Writes the core chunks for the given node to the given stream.

TBaseVirtualTree Class

AbsolutePath
Reads the overall index of a node.

AddChild
Creates and adds a new child node to given node.

AddFromStream
Adds the content from the given stream to the given node.

AddToSelection
Adds one or more nodes to the current selection.

AdjustPaintCellRect
Used in descendants to modify the clip rectangle of the current column while painting a certain node.

AdjustPanningCursor
Loads the proper cursor which indicates into which direction scrolling is done.
AdviseChangeEvent
Used to register a delayed change event.

AllocateInternalDataArea
Registration method to allocate tree internal data per node.

Animate
Support method for animated actions in the tree view.

Assign
Used to copy properties from another Virtual Treeview.

BeginDrag
Starts an OLE drag'n drop operation.

BeginSynch
Enters the tree into a special synchronized mode.

BeginUpdate
Locks the tree view to perform several update operations.

CalculateSelectionRect
Support method for draw selection.

CanAutoScroll
Determines whether the tree can currently auto scroll its window.

CancelCutOrCopy
Cancels any pending cut or copy clipboard operation.

CancelEditNode
Cancel the current edit operation, if there is any.

CanEdit
Determines whether a node can be edited or not.

CanFocus
Support method to determine whether the tree window can receive the input focus.

CanShowDragImage
Determines whether a drag image should be shown.

Change
Central method called when a node's selection state changes.

ChangeScale
Helper method called by the VCL when control resizing is due.

CheckParentCheckState
Helper method for recursive check state changes.
Clear
Clears the tree and removes all nodes.
ClearChecked
Not documented.
ClearSelection
Removes all nodes from the current selection.
ClearTempCache
Helper method to clear the internal temporary node cache.
ColumnIsEmpty
Used to determine if a cell is considered as being empty.
CopyTo
Copies Source and all its child nodes to Target.
CopyToClipBoard
Copies all currently selected nodes to the clipboard.
CountLevelDifference
Determines the level difference of two nodes.
CountVisibleChildren
Determines the number of visible child nodes of the given node.
Create
Constructor of the control
CreateParams
Prepares the creation of the controls window handle.
CreateWnd
Initializes data, which depends on the window handle.
CutToClipBoard
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.
DefineProperties
Helper method to customize loading and saving persistent tree data.
DeleteChildren
Removes all child nodes from the given node.
DeleteNode
Removes the given node from the tree.
DeleteSelectedNodes
Removes all currently selected nodes form the tree.
**Destroy**
Destructor of the control.

**DetermineHiddenChildrenFlag**
Determines whether all children of a given node are hidden.

**DetermineHiddenChildrenFlagAllNodes**
Determines whether all children of all nodes are hidden.

**DetermineHitPositionLTR**
Determines the hit position within a node with left-to-right and right-to-left orientation.

**DetermineHitPositionRTL**
Determines the hit position within a node with left-to-right and right-to-left orientation.

**DetermineNextCheckState**
Not documented.

**DetermineScrollDirections**
Not documented.

**DoAdvancedHeaderDraw**
Not documented.

**DoAfterCellPaint**
Not documented.

**DoAfterItemErase**
Not documented.

**DoAfterItemPaint**
Not documented.

**DoAfterPaint**
Not documented.

**DoAutoScroll**
Enables or disables the auto scroll timer.

**DoBeforeCellPaint**
Not documented.

**DoBeforeDrag**
Not documented.

**DoBeforeItemErase**
Not documented.

**DoBeforeItemPaint**
Not documented.

- **DoBeforePaint**
  Not documented.

- **DoCancelEdit**
  Called when the tree should stop editing without accepting changed values.

- **DoCanEdit**
  Not documented.

- **DoChange**
  Not documented.

- **DoCheckClick**
  Not documented.

- **DoChecked**
  Not documented.

- **DoChecking**
  Not documented.

- **DoChecked**
  Not documented.

- **DoCollapsing**
  Not documented.

- **DoColumnClick**
  Not documented.

- **DoColumnDblClick**
  Not documented.

- **DoColumnResize**
  Not documented.

- **DoCompare**
  Not documented.

- **DoCreateDataObject**
  Not documented.

- **DoCreateDragManager**
  Not documented.

- **DoCreateEditor**
  Not documented.

- **DoDragDrop**
Not documented.

- **DoDragExpand**
  Not documented.

- **DoDragging**
  Internal method which handles drag' drop.

- **DoDragOver**
  Not documented.

- **DoEdit**
  Initiates editing of the currently set focused column and edit node.

- **DoEndDrag**
  Not documented.

- **DoEndEdit**
  Stops the current edit operation and takes over the new content.

- **DoExpanded**
  Not documented.

- **DoExpanding**
  Not documented.

- **DoFocusChange**
  Not documented.

- **DoFocusChanging**
  Not documented.

- **DoFocusNode**
  Internal method to set the focused node.

- **DoFreeNode**
  Not documented.

- **DoGetAnimationType**
  Determines the type of animation to be used.

- **DoGetCursor**
  Not documented.

- **DoGetHeaderCursor**
  Not documented.

- **DoGetImageIndex**
  Not documented.

- **DoGetLineStyle**
  Not documented.
- **DoGetNodeHint**
  Not documented.

- **DoGetNodeTooltip**
  Not documented.

- **DoGetNodeWidth**
  Overridable method which always returns 0.

- **DoGetPopupMenu**
  Overridable method which triggers the OnGetPopup event.

- **DoGetUserClipboardFormats**
  Not documented.

- **DoHeaderClick**
  Not documented.

- **DoHeaderDoubleClick**
  Not documented.

- **DoHeaderDragged**
  Not documented.

- **DoHeaderDraggedOut**
  Not documented.

- **DoHeaderDragging**
  Not documented.

- **DoHeaderDraw**
  Not documented.

- **DoHeaderDrawQueryElements**
  Not documented.

- **DoHeaderMouseDown**
  Not documented.

- **DoHeaderMouseMove**
  Not documented.

- **DoHeaderMouseUp**
  Not documented.

- **DoHotChange**
  Not documented.

- **DoIncrementalSearch**
  Not documented.

- **DoInitChildren**
Not documented.

DoInitNode
Not documented.

DoKeyAction
Not documented.

DoLoadUserData
Not documented.

DoMeasureItem
Not documented.

DoNodeCopied
Not documented.

DoNodeCopying
Not documented.

DoNodeMoved
Not documented.

DoNodeMoving
Not documented.

DoPaintBackground
Not documented.

DoPaintDropMark
Overridable method which draws the small line on top of a node's image depending on the current drop state.

DoPaintNode
Overridable method which does nothing.

DoPopupMenu
Overridable method which shows the popup menu for the given node.

DoRenderOLEData
Not documented.

DoReset
Not documented.

DoSaveUserData
Not documented.

DoScroll
Overridable method which triggers the OnScroll event.

DoSetOffsetXY
Internal core routine to set the tree's scroll position.

- **DoShowScrollbar**
  Not documented.

- **DoStartDrag**
  Not documented.

- **DoStateChange**
  Not documented.

- **DoStructureChange**
  Not documented.

- **DoTimerScroll**
  Callback method which is triggered whenever the scroll timer fires.

- **DoUpdating**
  Not documented.

- **DoValidateCache**
  Not documented.

- **DragCanceled**
  Called by the VCL when a drag'n drop operation was canceled by the user.

- **DragDrop**
  Helper method, which is used when a drag operation is finished.

- **DragEnter**
  Not documented.

- **DragFinished**
  Called when a drag operation is finished (accepted or cancelled).

- **Dragging**
  Returns true if a drag'n drop operation is in progress.

- **DragLeave**
  Not documented.

- **DragOver**
  Not documented.

- **DrawDottedHLine**
  Not documented.

- **DrawDottedVLine**
  Not documented.

- **EditNode**
Starts editing the given node if allowed to.

- **EndEditNode**
  Stops node editing if it was started before.

- **EndSynch**
  Counterpart to BeginSynch.

- **EndUpdate**
  Resets the update lock set by BeginUpdate.

- **ExecuteAction**
  Not documented.

- **FindNodeInSelection**
  Helper method to find the given node in the current selection.

- **FinishChunkHeader**
  Not documented.

- **FinishCutOrCopy**
  Stops any pending cut or copy clipboard operation.

- **FlushClipboard**
  Renders all pending clipboard data.

- **FontChanged**
  Not documented.

- **FullCollapse**
  Collapses all nodes in the tree.

- **FullExpand**
  Expands all nodes in the tree.

- **GetBorderDimensions**
  Not documented.

- **GetCheckImage**
  Not documented.

- **GetCheckImageListFor**
  Not documented.

- **GetColumnClass**
  Returns the class to be used to manage columns in the tree.

- **GetControlsAlignment**
  Not documented.

- **GetDisplayRect**
  Returns the visible region used by the given node in client
coordinates.

- **GetFirst**
  Group of node navigation functions.

- **GetFirstChild**
  Group of node navigation functions.

- **GetFirstCutCopy**
  Group of node navigation functions.

- **GetFirstInitialized**
  Group of node navigation functions.

- **GetFirstNoInit**
  Group of node navigation functions.

- **GetFirstSelected**
  Group of node navigation functions.

- **GetFirstVisible**
  Group of node navigation functions.

- **GetFirstVisibleChild**
  Group of node navigation functions.

- **GetFirstVisibleChildNoInit**
  Group of node navigation functions.

- **GetFirstVisibleNoInit**
  Group of node navigation functions.

- **GetHeaderClass**
  Returns the header class to be used by the tree.

- **GetHitTestInfoAt**
  Returns information about the node at the given position.

- **GetImageIndex**
  Not documented.

- **GetLast**
  Group of node navigation functions.

- **GetLastChild**
  Group of node navigation functions.
GetLastChildNoInit
Group of node navigation functions.

GetLastInitialized
Group of node navigation functions.

GetLastNoInit
Group of node navigation functions.

GetLastVisible
Group of node navigation functions.

GetLastVisibleChild
Group of node navigation functions.

GetLastVisibleChildNoInit
Group of node navigation functions.

GetLastVisibleNoInit
Group of node navigation functions.

GetMaxColumnWidth
Returns the width of the largest node in the given column.

GetMaxRightExtend
Determines the maximum width of the currently visible part of the tree.

GetNativeClipboardFormats
Used to let descendants and the application add their own supported clipboard formats.

GetNext
Group of node navigation functions.

GetNextChecked
Not documented.

GetNextCutCopy
Group of node navigation functions.

GetNextInitialized
Group of node navigation functions.

GetNextNoInit
Group of node navigation functions.

GetNextSelected
Group of node navigation functions.

GetNextSibling
Group of node navigation functions.
- **GetNextVisible**
  Group of node navigation functions.
- **GetNextVisibleNoInit**
  Group of node navigation functions.
- **GetNextVisibleSibling**
  Group of node navigation functions.
- **GetNextVisibleSiblingNoInit**
  Group of node navigation functions.
- **GetNodeAt**
  Not documented.
- **GetNodeData**
  Returns the address of the user data area of the given node.
- **GetNodeLevel**
  Returns the indentation level of the given node.
- **GetOptionsClass**
  Customization helper to determine which options class the tree should use.
- **GetPrevious**
  Group of node navigation functions.
- **GetPreviousInitialized**
  Group of node navigation functions.
- **GetPreviousNoInit**
  Group of node navigation functions.
- **GetPreviousSibling**
  Group of node navigation functions.
- **GetPreviousVisible**
  Group of node navigation functions.
- **GetPreviousVisibleNoInit**
  Group of node navigation functions.
- **GetPreviousVisibleSibling**
  Group of node navigation functions.
- **GetPreviousVisibleSiblingNoInit**
  Group of node navigation functions.
- **GetSortedCutCopySet**
  Returns a sorted list of nodes, which are marked for cut or copy
clipboard operation.

- **GetSortedSelection**
  Returns a sorted list of all currently selected nodes.

- **GetTextInfo**
  Helper method for node editors, hints etc.

- **GetTreeFromDataObject**
  OLE drag'n drop and clipboard support method.

- **GetTreeRect**
  Returns the size of the virtual tree image.

- **GetVisibleParent**
  Returns the first (nearest) parent node, which is visible.

- **HandleHotTrack**
  Not documented.

- **HandleIncrementalSearch**
  Not documented.

- **HandleMouseDblClick**
  Not documented.

- **HandleMouseDown**
  Not documented.

- **HandleMouseUp**
  Not documented.

- **HasAsParent**
  Determines if the given node has got another node as one of its parents.

- **HasImage**
  Not documented.

- **HasPopupMenu**
  Determines whether there is a pop up menu assigned to the tree.

- **InitChildren**
  Not documented.

- **InitNode**
  Not documented.

- **InsertNode**
  Inserts a new node and returns it to the caller.

- **InternalAddFromStream**
Not documented.

- **InternalAddToSelection**
  Not documented.

- **InternalCacheNode**
  Not documented.

- **InternalClearSelection**
  Not documented.

- **InternalConnectNode**
  Not documented.

- **InternalData**
  Returns the address of the internal data for a tree class.

- **InternalDisconnectNode**
  Not documented.

- **InternalRemoveFromSelection**
  Not documented.

- **InvalidateCache**
  Empties the internal node cache and marks it as invalid.

- **InvalidateChildren**
  Invalidates all children of the given node.

- **InvalidateColumn**
  Invalidates the client area part of a column.

- **InvalidateNode**
  Invalidates the given node.

- **InvalidateToBottom**
  Invalidates the client area starting with the top position of the given node.

- **InvertSelection**
  Inverts the current selection.

- **IsEditing**
  Tells the caller whether the tree is currently in edit mode.

- **IsMouseSelecting**
  Tell the caller whether the tree is currently in draw selection mode.

- **IterateSubtree**
  Iterator method to go through all nodes of a given sub tree.

- **Loaded**
LoadFromFile
Loads previously streamed out tree data back in again.

LoadFromStream
Loads previously streamed out tree data back in again.

MainColumnChanged
Not documented.

MarkCutCopyNodes
Not documented.

MeasureItemHeight
Not documented.

MouseMove
Not documented.

MoveTo
Moves **Source** and all its child nodes to **Target**.

Notification
Not documented.

OriginalWMNCPaint
Not documented.

Paint
TControl's Paint method used here to display the tree.

PaintCheckImage
Not documented.

PaintImage
Not documented.

PaintNodeButton
Not documented.

PaintSelectionRectangle
Not documented.

PaintTree
Main paint routine for the tree image.

PaintTreeLines
Not documented.

PanningWindowProc
Not documented.
- **PasteFromClipboard**
  Inserts the content of the clipboard into the tree.
- **PrepareDragImage**
  Not documented.
- **Print**
  Not documented.
- **ProcessDrop**
  Helper method to ease OLE drag'n drop operations.
- **ProcessOLEData**
  Takes serialized OLE tree data and reconstructs the former structure.
- **ReadChunk**
  Not documented.
- **ReadNode**
  Not documented.
- **RedirectFontChangeEvent**
  Not documented.
- **ReinitChildren**
  Forces all child nodes of Node to be reinitialized.
- **ReinitNode**
  Forces a reinitialization of the given node.
- **RemoveFromSelection**
  Removes the given node from the current selection.
- **RenderOLEData**
  Renders pending OLE data.
- **RepaintNode**
  Causes the treeview to repaint the given node.
- **ResetNode**
  Resets the given node to uninitialized.
- **ResetRangeAnchor**
  Not documented.
- **RestoreFontChangeEvent**
  Not documented.
- **SaveToFile**
  Saves the entire content of the tree into a file or stream.
- **SaveToStream**
Saves the entire content of the tree into a file or stream.

- **ScrollIntoView**
  Scrolls the tree so that the given node comes in the client area.

- **SelectAll**
  Selects all nodes in the tree.

- **SelectNodes**
  Selects a range of nodes.

- **SetBiDiMode**
  Not documented.

- **SetFocusedNodeAndColumn**
  Not documented.

- **SkipNode**
  Not documented.

- **Sort**
  Sorts the given node.

- **SortTree**
  Sorts the entire tree view.

- **StartWheelPanning**
  Not documented.

- **StopWheelPanning**
  Not documented.

- **StructureChange**
  Not documented.

- **SuggestDropEffect**
  Not documented.

- **ToggleNode**
  Changes a node's expand state to the opposite state.

- **ToggleSelection**
  Toggles the selection state of a range of nodes.

- **UnselectAll**
  Deselects a range of nodes.

- **UpdateAction**
  Not documented.

- **UpdateDesigner**
  Not documented.
UpdateEditBounds
 Not documented.

UpdateHeaderRect
 Not documented.

UpdateHorizontalScrollBar
 Applies changes to the horizontal and vertical scrollbars.

UpdateScrollBars
 Applies changes to the horizontal and vertical scrollbars.

UpdateVerticalScrollBar
 Applies changes to the horizontal and vertical scrollbars.

UpdateWindowAndDragImage
 Not documented.

UseRightToLeftReading
 Helper method for right-to-left layout.

ValidateCache
 Initiates the validation of the internal node cache.

ValidateChildren
 Validates all children of a given node.

ValidateNode
 Validates a given node.

ValidateNodeDataSize
 Helper method for node data size initialization.

WndProc
 Redirected window procedure to do some special processing.

WriteChunks
 Writes the core chunks for the given node to the given stream.

WriteNode
 Writes the cover (envelop) chunk for the given node to the given stream.

**Legend**

- published

Property
Class Hierarchy

File
VirtualTrees

Links
Events, Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
**TVirtualStringTree.Action Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property Action;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.Align Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property Align;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.Alignment Property**

**TVirtualStringTree Class**

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

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.anchors Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property Anchors;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualStringTree Class**

**Links**

**TVirtualStringTree Class**

What do you think about this topic? Send feedback!
TVirtualStringTree.AnimationDuration Property

Determine the maximum duration the tree can use to play an animation.

Pascal

```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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.AutoExpandDelay Property**

*TVirtualStringTree Class*

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*

**Links**

*TVirtualStringTree Class*

*What do you think about this topic? Send feedback!*
TVirtualStringTree.AutoScrollDelay

**Property**

TVirtualStringTree Class

Time which determines when auto scrolling should start.

**Pascal**

```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

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.AutoScrollInterval Property**

**TVirtualStringTree Class**

Time interval between scroll events when doing auto scroll.

**Pascal**

```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

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.Background Property**  

**TVirtualStringTree Class**

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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree::BackgroundOffsetX Property**

*TVirtualStringTree Class*

Horizontal offset of the background image.

**Pascal**

```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*

**Links**

*TVirtualStringTree Class*
**TVirtualStringTree.BackgroundOffsetY Property**

**TVirtualStringTree Class**

Vertical offset of the background image.

**Pascal**

```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*

**Links**

*TVirtualStringTree Class*

What do you think about this topic? Send feedback!
TVirtualStringTree.BevelEdges Property

TVirtualStringTree Class

Not documented.

Pascal

```pascal
property BevelEdges;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.BevelInner Property**

**TVirtualStringTree Class**

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

**Links**

TVirtualStringTree Class

---

*What do you think about this topic?* Send feedback!
**TVirtualStringTree.BevelKind Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property BevelKind;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualStringTree Class**

**Links**

**TVirtualStringTree Class**

---

*What do you think about this topic? Send feedback!*
TVirtualStringTree.BevelOuter Property

TVirtualStringTree Class

Not documented.

Pascal

```pascal
property BevelOuter;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.BevelWidth Property
TVirtualStringTree Class

Not documented.

Pascal

```pascal
property BevelWidth;
```

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Class
TVirtualStringTree Class

Links
TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.BiDiMode Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property BiDiMode;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
TVirtualStringTree.BorderStyle Property

Same as TForm.BorderStyle.

Pascal

```pascal
property BorderStyle: TBorderStyle;
```

Description

See TForm.BorderStyle.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.BorderWidth Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property BorderWidth;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

---

What do you think about this topic? Send feedback!
**TVirtualStringTree.ButtonFillMode Property**

**TVirtualStringTree Class**

Determines how to fill the background of the node buttons.

**Pascal**

```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**

**Links**

**TVirtualStringTree Class**

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.ButtonStyle Property**

**TVirtualStringTree Class**

Determines the look of node buttons.

**Pascal**

```pascal
property ButtonStyle: TVTButtonStyle;
```

**Description**

Determines the look of node buttons.

**Class**

**TVirtualStringTree Class**

**Links**

**TVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.Canvas Property**

*TVirtualStringTree Class*

Not documented.

**Pascal**

```pascal
property Canvas;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

*TVirtualStringTree Class*

**Links**

*TVirtualStringTree Class*

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.ChangeDelay Property**

**TVirtualStringTree Class**

Time which determines when the OnChangeEvent 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 OnChangeEvent should occur. A value of 0 means to trigger OnChangeEvent immediately after the change (usually a selection or focus change) happen. Any value > 0 will start a timer which then triggers OnChangeEvent.

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

**Links**

TVirtualStringTree Class
What do you think about this topic? Send feedback!
**TVirtualStringTree.CheckImageKind Property**

**TVirtualStringTree Class**

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**

**Links**

**TVirtualStringTree Class**

*What do you think about this topic? Send feedback!***
TVirtualStringTree.ClipboardFormats Property

Special class to keep a list of clipboard format descriptions.

Pascal

```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.
**TVirtualStringTree.Color Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property Color;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.Colors Property**

*TVirtualStringTree Class*

A collection of colors used in the tree.

**Pascal**

```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*

**Links**

*TVirtualStringTree Class*

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.Constraints Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property Constraints;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualStringTree Class**

**Links**

**TVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.Ctl3D Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property Ctl3D;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

---

*What do you think about this topic?* Send feedback!
**TVirtualStringTree.CustomCheckImages Property**

**TVirtualStringTree Class | See Also**

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

**Links**

TVirtualStringTree Class, See Also
**TVirtualStringTree.DefaultNodeHeight Property**

**TVirtualStringTree Class**

Read or set the height new nodes get as initial value.

**Pascal**

```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**

**Links**

**TVirtualStringTree Class**

---

*What do you think about this topic? Send feedback!*
TVirtualStringTree.DefaultPasteMode Property

Read or set the value, which determines where to add pasted nodes to.

Pascal

```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

Links

TVirtualStringTree Class, See Also
What do you think about this topic? Send feedback!
**TVirtualStringTree.DefaultText Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```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**

**Links**

**TVirtualStringTree Class**

*What do you think about this topic?*  *Send feedback!*
**TVirtualStringTree.DragCursor Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```plaintext
property DragCursor;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

---

What do you think about this topic? Send feedback!
**TVirtualStringTree.DragHeight Property**

**TVirtualStringTree Class**

Read or set the vertical limit of the internal drag image.

**Pascal**

```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

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.DragImageKind Property**

*TVirtualStringTree Class*

Read or set what should be shown in the drag image.

**Pascal**

```pascal
property DragImageKind: TVTDragImageKind;
```

**Description**

DragImageKind allows to switch parts of the drag image off and on.

**Class**

*TVirtualStringTree Class*

**Links**

*TVirtualStringTree Class*

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.DragKind Property**

**TVirtualStringTree Class**

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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.DragMode Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property DragMode;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TVirtualStringTree.DragOperations Property

TVirtualStringTree Class

Read or set which drag operations may be allowed in the tree.

Pascal

```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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.DragType Property**

**TVirtualStringTree Class**

Read or set which subsystem should be used for dragging.

**Pascal**

```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**

**Links**

**TVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.DragWidth Property**

**TVirtualStringTree Class**

Read or set the horizontal limit of the internal drag image.

**Pascal**

```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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.DrawSelectionMode Property**

Read or set how multiselection with the mouse is to be visualized.

**Pascal**

```pascal
property DrawSelectionMode: TVTDrawSelectionMode;
```

**Description**

Virtuall Treeview allows to display two different selection rectangles when doing multiselection with the mouse. One is the traditional 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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TVirtualStringTree/EditDelay Property

TVirtualStringTree Class | See Also

Read or set the maximum time between two single clicks on the same node, which should start node editing.

Pascal

```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 maximum time distance between both clicks in which the edit operation is started.

See Also

Editors and editing

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualStringTree.Enabled Property

TVirtualStringTree Class

Not documented.

Pascal

```pascal
property Enabled;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.Font Property**

**TVirtualStringTree Class**

Same as TWinControl.Font.

**Pascal**

```pascal
property Font;
```

**Description**

See TWinControl.Font.

**Class**

 TVirtualStringTree Class

**Links**

 TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.Header Property**

**TVirtualStringTree Class | See Also**

Provides access to the header instance.

**Pascal**

```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

**Links**

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualStringTree.HintAnimation Property

TVirtualStringTree Class

Read or set the current hint animation type.

Pascal

```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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.HintMode Property**

**TVirtualStringTree Class**

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

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
TVirtualStringTree.HotCursor Property

TVirtualStringTree Class | See Also

Read or set which cursor should be used for hot nodes.

Pascal

```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 whatever you like. The HotCursor property is used to specify, which cursor is to be used.

See Also

HotNode, TVTPaintOptions

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class, See Also
What do you think about this topic? Send feedback!
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

Links

TVirtualStringTree Class, See Also
**TVirtualStringTree.IncrementalSearch Property**

**TVirtualStringTree Class | See Also**

Read or set the current incremental search mode.

**Pascal**

```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

**Links**

TVirtualStringTree Class, See Also
**TVirtualStringTree.IncrementalSearchDirection Property**

Read or set the direction to be used for incremental search.

**Pascal**

```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, IncrementalSearchTime

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class, See Also
TVirtualStringTree.IncrementalSearchStart Property

TVirtualStringTree Class | See Also

Read or set where to start incremental search.

Pascal

```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

Links

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualStringTree.IncrementalSearchTimeout Property

TVirtualStringTree Class | See Also

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

Links
TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualStringTree.Indent Property

TVirtualStringTree Class

Read or set the indentation amount for node levels.

Pascal

```pascal
property Indent: Cardinal;
```

Description

Each new level in the tree (child nodes of a parent node) are visually shifted to distinguish between them and their parent node (that's the tree layout after all). The Indent property determines the shift distance in pixels.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.LineMode** Property

**TVirtualStringTree Class**

Read or set the mode of the tree lines.

**Pascal**

```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**

**Links**

**TVirtualStringTree Class**

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.LineStyle Property**

**TVirtualStringTree Class**

Read or set the mode of the tree lines.

**Pascal**

```
propertyLineStyle: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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.Margin Property**

*TVirtualStringTree Class | See Also*

Read or set the tree's node margin.

**Pascal**

```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

**Links**
TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualStringTree.NodeAlignment Property**

TVirtualStringTree Class | See Also

Read or set the node alignment value.

**Pascal**

```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

**Links**

TVirtualStringTree Class, See Also

*What do you think about this topic? Send feedback!*
TVirtualStringTree.NodeDataSize Property

TVirtualStringTree Class | See Also

Read or set the extra data size for each node.

Pascal

```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

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualStringTree.OnAdvancedHeaderDraw

**Event**

TVirtualStringTree Class | See Also

Header paint support event.

**Pascal**

```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

Links
TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualStringTree.OnAfterCellPaint Event

Paint support event.

Pascal

```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

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class, See Also
TVirtualStringTree.OnAfterItemErase Event

Paint support event.

Pascal

```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

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnAfterItemPaint Event**

Paint support event.

**Pascal**

```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

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class, See Also

---

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnAfterPaint Event**

**TVirtualStringTree Class | See Also**

Paint support event.

**Pascal**

```
property OnAfterPaint: TVPaintEvent;
```

**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

**Class**

- TVirtualStringTree Class

**Links**

- TVirtualStringTree Class, See Also

---

*What do you think about this topic? Send feedback!*
TVirtualStringTree.OnBeforeCellPaint Event

Paint support event.

Pascal

```pascal
property OnBeforeCellPaint: TVTBeforeCellPaintEvent;
```

Description

This event is called immediately before a cell is painted.

See Also

Paint cycles and stages

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class, See Also
**TVirtualStringTree.OnBeforeItemErase Event**

**TVirtualStringTree Class | See Also**

Paint support event.

**Pascal**

```pascal
property OnBeforeItemErase: TVTBeforeItemEraseEvent;
```

**Description**

Called when the background of a node is about to be erased.

**See Also**

Paint cycles and stages

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnBeforeItemPaint Event**

*TVirtualStringTree Class | See Also*

Paint support event.

**Pascal**

```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

**Class**

*TVirtualStringTree Class*

**Links**

*TVirtualStringTree Class, See Also*

---

What do you think about this topic? Send feedback!
TVirtualStringTree.OnBeforePaint Event

TVirtualStringTree Class | See Also

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

Class

- TVirtualStringTree Class

Links

- TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnChangeEvent Event**

**TVirtualStringTree Class**

- Navigation support event.

**Pascal**

```pascal
propertyOnChange:TVTChangeEvent;
```

**Description**

Called when a node's selection state has changed.

**Class**

- TVirtualStringTree Class

**Links**

- TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnChecked Event**

**TVirtualStringTree Class**

Check support event.

**Pascal**

```
property OnChecked: TVTChangeEvent;
```

**Description**

Triggered when a node's check state has changed.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

---

*What do you think about this topic?* Send feedback!
**TVirtualStringTree.OnChecking Event**

**TVirtualStringTree Class**

Check support event.

**Pascal**

```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

**Links**

TVirtualStringTree Class

---

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnClick Property**

*TVirtualStringTree Class*

Not documented.

**Pascal**

```pascal
property OnClick;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

*TVirtualStringTree Class*

**Links**

*TVirtualStringTree Class*

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnCollapsed Event**

**TVirtualStringTree Class**

Miscellaneous event.

**Pascal**

```pascal
property OnCollapsed: TVTChangeEvent;
```

**Description**

Triggered after a node has been collapsed, that is, its child nodes are no longer displayed.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnCollapsing Event**

**TVirtualStringTree Class**

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

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnColumnButtonClick Event**

**TVirtualStringTree Class | See Also**

Header and column support event.

**Pascal**

```
property OnColumnButtonClick: TVTColumnButtonClickEvent;
```

**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

**Links**

TVirtualStringTree Class, See Also

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree. OnColumnDblClick Event**

**TVirtualStringTree Class | See Also**

Header and column support event.

**Pascal**

```pascal
property OnColumnDblClick: TVTColumnDblClickEvent;
```

**Description**

Same as OnColumnClick but for double clicks.

**See Also**

OnColumnClick, OnHeaderDblClick

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class, See Also

*What do you think about this topic? Send feedback!*
TVirtualStringTree.OnColumnResize Event

TVirtualStringTree Class

Header and column support routine.

Pascal

```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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.OnCompareNodes Event

TVirtualStringTree Class | See Also

Sort and search support event.

Pascal

```pascal
property OnCompareNodes: TVTCompareEvent;
```

Description

This event is the core event for all comparisons 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 are 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 comparison result as would there be an ascending sort order.

Below is some sample code taken from the Advanced Demo:
procedure TMainForm.VDT1CompareNodes(Sender: TBaseVirtualTree;
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

Links
TVirtualStringTree Class, See Also
**TVirtualStringTree.OnCreateDataObject Event**

**TVirtualStringTree Class**

Drag'n drop support event.

**Pascal**

```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. Descendants (which override DoGetDataObject) or the application can return an own IDataObject implementation to support special formats.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TVirtualStringTree.OnCreateDragManager Event

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

Links
TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.OnCreateEditor Event

TVirtualStringTree Class | See Also

Editing support event.

Pascal

```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. Descendants like TCustomVirtualStringTree supply a generic and simple string editor.

See Also

Editors and editing

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnDbClick Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property OnDbClick;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TVirtualStringTree.OnDragAllowed Event

TVirtualStringTree Class

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 preferred over doing a manual BeginDrag.

Notes

The OnDragAllowed event is called only if the current DragMode is dmManual.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnDragDrop Event**

**TVirtualStringTree Class**

Drag'n drop support event.

**Pascal**

```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_VIRTUALTREETREE 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_VIRTUALTREETREE 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.

```pascal
procedure TMainForm.VTDragDrop(Sender: TBaseVirtualTree;
const Formats: array of Word;
Shift: TShiftState;
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 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
        AttachMode := amInsertBefore
      else
        AttachMode := amNowhere;
    end;
    // in the case the drop target does an optimized move effect
    // to indicate this also to the drag source
    Sender.ProcessDrop(DataObject, Sender.DropTargetNode, Effect, AttachMode);
    Break;
  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

Links
TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnDragOver Event**

*TVirtualStringTree Class | See Also*

Drag'n drop support event.

**Pascal**

```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*

**Links**

*TVirtualStringTree Class, See Also*
**TVirtualStringTree.OnEditCancelled Event**

**TVirtualStringTree Class | See Also**

Editing support event.

**Pascal**

```pascal
property OnEditCancelled: TVTEditCancelEvent;
```

**Description**

Triggered when an edit action has been cancelled.

**See Also**

Editors and editing

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class, See Also

---

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnEdited Event**

*TVirtualStringTree Class | See Also*

Editing support event.

**Pascal**

```pascal
property OnEdited: TVTEditChangeEvent;
```

**Description**

Triggered when an edit action has successfully been finished.

**See Also**

Editors and editing

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class, See Also

*What do you think about this topic? Send feedback!*
TVirtualStringTree.OnEditing Event

Editing support event.

Pascal

```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

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class, See Also
TVirtualStringTree.OnEndDock Property

TVirtualStringTree Class

Not documented.

Pascal

```pascal
property OnEndDock;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnEndDrag Property**

**TVirtualStringTree Class**

Not documented.

Pascal

```pascal
property OnEndDrag;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnEnter Property**

**TVirtualStringTree Class**

Not documented.

Pascal

```pascal
property OnEnter;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TVirtualStringTree. OnExit Property

TVirtualStringTree Class

Not documented.

Pascal

```pascal
property OnExit;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnExpanded Event**

**TVirtualStringTree Class**

Missellaneous event.

**Pascal**

```pascal
property OnExpanded: TVTChangeEvent;
```

**Description**

Triggered after a node has been expanded.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.OnExpanding Event

TVirtualStringTree Class

Miscellaneous event.

Pascal

```pascal
property OnExpanding: TVTChangingEvent;
```

Description
Triggered just before a node is expanded. Use **Allowed** to allow or deny this action.

Class
TVirtualStringTree Class

Links
TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.OnFocusChanged Event

TVirtualStringTree Class

Navigation support event.

Pascal

```pascal
property OnFocusChanged: TTVFocusChangeEvent;
```

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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnFocusChanging Event**

**TVirtualStringTree Class**

Navigation support event.

**Pascal**

```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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnFreeNode Event**

TVirtualStringTree Class

Data management node.

**Pascal**

```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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnGetCellIsEmpty Event**

**TVirtualStringTree Class**

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

**Links**

TVirtualStringTree Class
What do you think about this topic? Send feedback!
**TVirtualStringTree.OnGetCursor Event**

**TVirtualStringTree Class**

Miscellaneous event.

**Pascal**

```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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnGetHeaderCursor Event**

**TVirtualStringTree Class**

Header and column support event.

**Pascal**

```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

**Links**

TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.OnGetHelpContext Event

TVirtualStringTree Class

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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.**ㅤ**OnGetHint**  Event

**TVirtualStringTree**  Class

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

**Links**

**TVirtualStringTree**  Class

---

*What do you think about this topic? Send feedback!*
`TVirtualStringTree.OnGetImageIndex` Event

`TVirtualStringTree` Class

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 they 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

Links
   TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnGetImageIndexEx Event**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnGetLineStyle Event**

*TVirtualStringTree Class | See Also*

Display management event.

**Pascal**

```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

Links
TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnGetNodeDataSize Event**

**TVirtualStringTree Class | See Also**

Data management event.

**Pascal**

```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

**Class**

TVirtualStringTree Class
What do you think about this topic? Send feedback!
**TVirtualStringTree.OnGetPopupMenu Event**

**TVirtualStringTree Class**

Miscellaneous event.

**Pascal**

```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

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
Virtual string tree event to query for a node's normal or static text.

Pascal

```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

Links
TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualStringTree.OnGetUserClipboardFormats Event

TVirtualStringTree Class

Drag'n drop and clipboard support event.

Pascal

```
property OnGetUserClipboardFormats: TVTGetUserClipboardFormats;
```

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 descendants (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**
Links
TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnHeaderClick Event**

**TVirtualStringTree Class | See Also**

Header & column support event.

**Pascal**

```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

**Links**

TVirtualStringTree Class, See Also
TVirtualStringTree.OnHeaderDblClick Event

TVirtualStringTree Class | See Also

Header & column support event.

Pascal

```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

Links

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualStringTree.OnHeaderDragged Event

TVirtualStringTree Class

Header & column support event.

Pascal

```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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.OnHeaderDraggedOut Event

TVirtualStringTree Class

Header & column support event.

Pascal

```pascal
property OnHeaderDraggedOut: TTVTHeaderDraggedOutEvent
```

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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnHeaderDragging Event**

**TVirtualStringTree Class**

Header & column support event.

**Pascal**

```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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnHeaderDraw Event**

**TVirtualStringTree Class**

Header & column support event.

**Pascal**

```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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TVirtualStringTree.OnHeaderDrawQueryElements Event

TVirtualStringTree Class | See Also

Header & column support event.

**Pascal**

```pascal
property OnHeaderDrawQueryElements: TVTHeaderPaintQueryElements;
```

**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

**Links**

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnHeaderMouseDown Event**

**TVirtualStringTree Class**

Header & column support event.

**Pascal**

```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**

**Links**

**TVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnHeaderMouseMove Event**

**TVirtualStringTree Class**

Header & column support event.

**Pascal**

```pascal
property OnHeaderMouseMove: TVTHeaderMouseMoveEvent;
```

**Description**

This event is triggered when the mouse pointer is moved over the header area.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnHeaderMouseUp**

**TVirtualStringTree Class**

Header & column support event.

**Pascal**

```pascal
property OnHeaderMouseUp: TVTHeaderMouseEvent;
```

**Description**

This event is very much like OnHeaderMouseDown but is triggered when a mouse button is released.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

---

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnHotChange Event**

**TVirtualStringTree Class**

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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnIncrementalSearch Event**

**TVirtualStringTree Class**

Miscellaneous event.

**Pascal**

```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.

```pascal
procedure TfrmProperties.VST3IncrementalSearch(Sender: TObject; var Result: Integer);
var
    S, PropText: string;
begin
    // Note: This code requires a proper Unicode/Wide char
```
// size and clarity reasons. For now strings are
// 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];
end;

// By using StrLIComp we can specify a maximum length
// 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

Links
TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.OnInitChildren Event

Node management event.

Pascal

```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 initialization without 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

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class, See Also
What do you think about this topic? Send feedback!
TVirtualStringTree.OnInitNode Event

TVirtualStringTree Class | See Also

Node management event.

Pascal

```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

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualStringTree.OnKeyAction Event

TVirtualStringTree Class

Miscellaneous event.

Pascal

```pascal
property OnKeyAction: TVTKeyActionEvent;
```

Description

This event is a convinient way for the application or descendant 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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnKeyDown Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property OnKeyDown;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

`TVirtualStringTree Class`

**Links**

`TVirtualStringTree Class`

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnKeyPress Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property OnKeyPress;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TVirtualStringTree.OnKeyUp Property

TVirtualStringTree Class

Not documented.

Pascal

```pascal
property OnKeyUp;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnLoadNode Event**

*TVirtualStringTree Class* | *See Also*

Streaming support event.

**Pascal**

```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*

**Links**

*TVirtualStringTree Class, See Also*
What do you think about this topic? Send feedback!
**TVirtualStringTree.OnMeasureItem Event**

**TVirtualStringTree Class | See Also**

Miscellaneous event.

**Pascal**

```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
Links

TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
TVirtualStringTree.OnMouseDown

Property
TVirtualStringTree Class

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

Links
TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnMouseMove Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property OnMouseMove;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualStringTree Class**

**Links**

**TVirtualStringTree Class**

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnMouseUp Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property OnMouseUp;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class
**TVirtualStringTree.OnMouseWheel Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property OnMouseWheel;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualStringTree Class**

**Links**

**TVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
TVirtualStringTree.OnNewText Event

Pascal

```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
**TVirtualStringTree.OnNodeCopied Event**

**TVirtualStringTree Class**

Miscellaneous event.

**Pascal**

```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**

**Links**

**TVirtualStringTree Class**

---

*What do you think about this topic? Send feedback!*
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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree. OnNodeMoved Event

TVirtualStringTree Class

Miscellaneous event.

Pascal

property OnNodeMoved: TVTNodeMovedEvent;

Description

This event is very much like OnNodeCopied but used for moving nodes instead.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnNodeMoving Event**

**TVirtualStringTree Class**

Miscellaneous event.

**Pascal**

```
property OnNodeMoving: TVTNodeMovingEvent;
```

**Description**

This event is very much like OnNodeCopying but used for moving nodes instead.

**Class**

**TVirtualStringTree Class**

**Links**

**TVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnPaintBackground Event**

**TVirtualStringTree Class**

Paint support event.

**Pascal**

```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

**Links**

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnPaintText Event**

**TVirtualStringTree Class | See Also**

Event to change text formatting for particular nodes.

**Pascal**

```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 recalcula\ntion of the node width by calling InvalidateNode (when the con\nditions for the changed formatting are met - not in the event han\ndler for OnPaintText).

See Also
Paint cycles and stages

Class
TVirtualStringTree Class

Links
TVirtualStringTree Class, See Also
TVirtualStringTree.OnRenderOLEData Event

TVirtualStringTree Class

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

Links

TVirtualStringTree Class
**TVirtualStringTree.OnResetNode Event**

**TVirtualStringTree Class | See Also**

Node management event.

**Pascal**

```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

**Links**

TVirtualStringTree Class, See Also

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnResize Property**

**TVirtualStringTree Class**

Not documented.

Pascal

```pascal
property OnResize;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualStringTree Class**

**Links**

**TVirtualStringTree Class**

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnSaveNode Event**

TVirtualStringTree Class | See Also

Streaming support event.

**Pascal**

```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

Links
TVirtualStringTree Class, See Also

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnScroll Event**

**TVirtualStringTree Class | See Also**

Miscellaneous event.

**Pascal**

```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

**Links**

TVirtualStringTree Class, See Also

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnShortenString Event**

**TVirtualStringTree Class**

String tree event for custom handling of string abbreviations.

**Pascal**

```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`

**Links**
TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnShowScrollbar Event**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnStartDock Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property OnStartDock;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.OnStartDrag Property**

**TVirtualStringTree Class**

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*

**Links**

*TVirtualStringTree Class*

*What do you think about this topic? Send feedback!*
TVirtualStringTree.OnStateChange Event

TVirtualStringTree Class

Miscellaneous event.

Pascal

```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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.OnStructureChange Event**

**TVirtualStringTree Class**

Miscellaneous event.

**Pascal**

```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

**Links**

TVirtualStringTree Class

---

What do you think about this topic? Send feedback!
TVirtualStringTree.OnUpdating Event

TVirtualStringTree Class

Miscellaneous event.

Pascal

```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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
**TVirtualStringTree.ParentBiDiMode Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property ParentBiDiMode;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.ParentColor Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property ParentColor;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.ParentCtl3D Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property ParentCtl3D;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.ParentFont Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property ParentFont;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualStringTree Class**

**Links**

**TVirtualStringTree Class**
**TVirtualStringTree.ParentShowHint Property**

**TVirtualStringTree Class**

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

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.PopupMenu Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property PopupMenu;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.RootNodeCount Property**

Read or set the number of nodes on the top level.

**Pascal**

```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`

**Links**

`TVirtualStringTree Class`

---

*What do you think about this topic? Send feedback!*
TVirtualStringTree.ScrollBarOptions

Property

TVirtualStringTree Class

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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.SelectionBlendFactor Property

TVirtualStringTree Class | See Also

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

Links
TVirtualStringTree Class, See Also
**TVirtualStringTree.SelectionCurveRadius Property**

**TVirtualStringTree Class | See Also**

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

**Links**

TVirtualStringTree Class, See Also

---

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.ShowHint Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property ShowHint;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.StateImages Property**

**TVirtualStringTree Class | See Also**

Reference to the images list which is used for the state images.

**Pascal**

```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

**Links**

TVirtualStringTree Class, See Also

*What do you think about this topic? Send feedback!*
**TVirtualStringTree.TabOrder Property**

**TVirtualStringTree Class**

Not documented.

**Pascal**

```pascal
property TabOrder;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic?* Send feedback!
TVirtualStringTree.TabStop Property

TVirtualStringTree Class

Not documented.

Pascal

```pascal
property TabStop;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic?  Send feedback!
**TVirtualStringTree.TextMargin Property**

Read or set the distance of the node caption to its borders.

**Pascal**

```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

**Links**

TVirtualStringTree Class, See Also
What do you think about this topic? Send feedback!
**TVirtualStringTree.TreeOptions Property**

**TVirtualStringTree Class**

Reference to the tree's options.

**Pascal**

```pascal
property TreeOptions: TStringTreeOptions;
```

**Description**

The tree options are one of the main switches to modify a treeview's behavior. Virtual Treeview supports customizing tree options by descendents. 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 descendants.

**Class**

TVirtualStringTree Class

**Links**

TVirtualStringTree Class

*What do you think about this topic? Send feedback!*
TVirtualStringTree.Visible Property

TVirtualStringTree Class

Not documented.

Pascal

```pascal
property Visible;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.WantTabs Property

TVirtualStringTree Class

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

Links

TVirtualStringTree Class

What do you think about this topic? Send feedback!
TVirtualStringTree.GetOptionsClass Method

TVirtualStringTree Class

Customization helper to determine which options class the tree should use.

Pascal

```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 GetOptionsClass;
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
  ...
end;
Code

```pascal
Result := TStringTreeOptions;
end;
```

Class

**TVirtualStringTree Class**

Links

**TVirtualStringTree Class**

---

*What do you think about this topic? Send feedback!*
TVirtualTreeColumn Class

Represents a column in a Virtual Treeview.

Pascal

```pascal
TVirtualTreeColumn = class(TCollectionItem);
```

Description

This enhanced collection item, which is organized within the TCollection descendant `TVirtualTreeColumns`, manages all aspects of a single column.

Group

Classes

Members

Properties

- Alignment
  Not documented.
- BiDiMode
  Not documented.
- Color
  Not documented.
- Hint
  Not documented.
- ImageIndex
  Not documented.
- Layout
Not documented.

- **Left**
  Not documented.

- **Margin**
  Not documented.

- **MaxWidth**
  Not documented.

- **MinWidth**
  Not documented.

- **Options**
  Not documented.

- **Owner**
  Not documented.

- **Position**
  Not documented.

- **Spacing**
  Not documented.

- **Style**
  Not documented.

- **Tag**
  Not documented.

- **Text**
  Not documented.

- **Width**
  Not documented.

**Methods**

- **Assign**
  Not documented.

- **ComputeHeaderLayout**
  Calculates the layout of a column header.

- **Create**
  Not documented.

- **DefineProperties**
Not documented.

- **Destroy**
  Not documented.
- **Equals**
  Not documented.
- **GetAbsoluteBounds**
  Not documented.
- **GetDisplayName**
  Not documented.
- **GetOwner**
  Not documented.
- **GetRect**
  Returns the rectangle this column occupies in the header (relative to (0, 0) of the non-client area).
- **LoadFromStream**
  Not documented.
- **ParentBiDiModeChanged**
  Not documented.
- **ParentColorChanged**
  Not documented.
- **ReadHint**
  Not documented.
- **ReadText**
  Not documented.
- **RestoreLastWidth**
  Not documented.
- **SaveToStream**
  Not documented.
- **UseRightToLeftReading**
  Not documented.
- **WriteHint**
  Not documented.
- **WriteText**
  Not documented.
Legend

- published
- Property
- public
- read only
- Method
- virtual
- protected

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.Alignment Property**

**TVirtualTreeColumn Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumn Class

*What do you think about this topic? Send feedback!*
TVirtualTreeColumn.BiDiMode Property
TVirtualTreeColumn Class

Not documented.

Pascal

```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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
TVirtualTreeColumn.Color Property

TVirtualTreeColumn Class

Not documented.

Pascal

```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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
TVirtualTreeColumn.Hint Property

TVirtualTreeColumn Class

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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.ImageIndex Property**

**TVirtualTreeColumn Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumn Class

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumn.Layout Property**

**TVirtualTreeColumn Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.Left Property**

**TVirtualTreeColumn Class**

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*

**Links**

*TVirtualTreeColumn Class*

---

_What do you think about this topic?_  Send feedback!
**TVirtualTreeColumn.Margin Property**

**TVirtualTreeColumn Class**

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

**Links**

TVirtualTreeColumn Class

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumn.MaxWidth Property**

**TVirtualTreeColumn Class**

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

**Links**

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
TVirtualTreeColumn.MinWidth Property
TVirtualTreeColumn Class

Not documented.

Pascal

```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

Links
TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
TVirtualTreeColumn.Options Property

TVirtualTreeColumn Class

Not documented.

Pascal

```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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.Owner Property**

**TVirtualTreeColumn Class**

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**

**Links**

**TVirtualTreeColumn Class**

*What do you think about this topic? Send feedback!*
TVirtualTreeColumn.Position Property

TVirtualTreeColumn Class

Not documented.

Pascal

```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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
TVirtualTreeColumn.Spacing Property

TVirtualTreeColumn Class

Not documented.

Pascal

```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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.Style Property**

**TVirtualTreeColumn Class**

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`

**Links**

`TVirtualTreeColumn Class`

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumn.Tag Property**

*TVirtualTreeColumn Class*

Not documented.

**Pascal**

```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*

**Links**

*TVirtualTreeColumn Class*
## TVirtualTreeColumn.Text Property

### TVirtualTreeColumn Class

Not documented.

### Pascal

```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

### Links

TVirtualTreeColumn Class

---

*What do you think about this topic? Send feedback!*
**TVirtualTreeNode.Width Property**

TVirtualTreeNode Class

Not documented.

**Pascal**

```pascal
property Width: Integer;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualTreeNode Class

**Links**

TVirtualTreeNode Class

*What do you think about this topic? Send feedback!*
TVirtualTreeColumn.Assign Method

TVirtualTreeColumn Class

Not documented.

Pascal

```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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
TVirtualTreeColumn.ComputeHeaderLayout Method

TVirtualTreeColumn Class

Calculates the layout of a column header.

Pascal

```pascal
procedure ComputeHeaderLayout(DC: HDC; const Client:
```

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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.Create Constructor**

**Class**

TVirtualTreeColumn Class

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualTreeColumn Class

**Links**

TVirtualTreeColumn Class

---

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.DefineProperties Method**
**TVirtualTreeColumn Class**

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**

**Links**

TVirtualTreeColumn Class

---

*What do you think about this topic? Send feedback!*
TVirtualTreeColumn.Destroy Destructor

TVirtualTreeColumn Class

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

Links
TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
TVirtualTreeColumn.Equals Method

TVirtualTreeColumn Class

Not documented.

Pascal

\begin{verbatim}
function Equals(OtherColumn: TVirtualTreeColumn): Boolean;
\end{verbatim}

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumn Class

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
TVirtualTreeColumn.GetAbsoluteBounds Method

TVirtualTreeColumn Class

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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.GetDisplayName Method**

**TVirtualTreeColumn Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.GetOwner Method**

**TVirtualTreeColumn Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.GetRect Method**

**TVirtualTreeColumn Class**

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

**Links**

TVirtualTreeColumn Class

*What do you think about this topic?* Send feedback!
**TVirtualTreeColumn.LoadFromStream Method**

**TVirtualTreeColumn Class**

Not documented.

**Pascal**

```
procedure LoadFromStream(const Stream: TStream; Vers
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualTreeColumn Class

**Links**

TVirtualTreeColumn Class

---

*What do you think about this topic? Send feedback!*
TVirtualTreeColumn.ParentBiDiModeChanged Method

TVirtualTreeColumn Class

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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.SetParentColorChanged Method**

*TVirtualTreeColumn Class*

Not documented.

**Pascal**

```pascal
procedure ParentColorChanged;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

*TVirtualTreeColumn Class*

**Links**

*TVirtualTreeColumn Class*
TVirtualTreeColumn.ReadHint Method

TVirtualTreeColumn Class

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

Links
TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.ReadText Method**

**TVirtualTreeColumn Class**

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

**Links**

TVirtualTreeColumn Class

---

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumn.RestoreLastWidth Method**

**TVirtualTreeColumn Class**

Not documented.

**Pascal**

```pascal
procedure RestoreLastWidth;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualTreeColumn Class**

**Links**

**TVirtualTreeColumn Class**

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.SaveToStream Method**

**TVirtualTreeColumn Class**

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**

**Links**

**TVirtualTreeColumn Class**

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumn.UseRightToLeftReading Method**

**TVirtualTreeColumn Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumn Class

*What do you think about this topic? Send feedback!*
TVirtualTreeColumn.WriteHint Method

TVirtualTreeColumn Class

Not documented.

Pascal

```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

Links

TVirtualTreeColumn Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumn.WriteText Method**

**TVirtualTreeColumn Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumn Class

---

*What do you think about this topic? Send feedback!*
TVirtualTreeColumns Class

Description
This class is an enhanced collection which manages general aspects of columns like ordering, traversal, streaming, painting, dragging etc.

Group
Classes

Members

Properties

- ClickIndex
  Not documented.
- Header
  Not documented.
- HeaderBitmap
  Not documented.
- Items
  Not documented.
- PositionToIndex
  Not documented.
- TrackIndex

Pascal

```pascal
TVirtualTreeColumns = class(TCollection);
```
Not documented.

Methods

- **Add**
  
  Not documented.

- **AdjustAutoSize**
  
  Called when columns must be sized so that the fit the client area.

- **AdjustDownColumn**
  
  Determines the column from the given position and returns it.

- **AdjustHoverColumn**
  
  Determines the new hover column index and returns true if the index actually changed else False.

- **AdjustPosition**
  
  Reorders the column position array so that the given column gets the given position.

- **AnimatedResize**
  
  Resizes the given column animated by scrolling the window DC.

- **Assign**
  
  Not documented.

- **Clear**
  
  Not documented.

- **ColumnFromPosition**
  
  Returns the index of the column at the given position.

- **Create**
  
  Not documented.

- **Destroy**
  
  Not documented.

- **DrawButtonText**
  
  Not documented.

- **DrawXPButton**
  
  Helper procedure to draw an Windows XP like header button.

- **Equals**
  
  Compares itself with the given set of columns.

- **FixPositions**
Fixes column positions after loading from DFM.

- **GetColumnAndBounds**
  Returns the column where the mouse is currently in as well as the left and right bound of this column.

- **GetColumnBounds**
  Returns the left and right bound of the given column.

- **GetFirstVisibleColumn**
  Returns the index of the first visible column or "InvalidColumn" if either no columns are defined or all columns are hidden.

- **GetLastVisibleColumn**
  Returns the index of the last visible column or "InvalidColumn" if either no columns are defined or all columns are hidden.

- **GetNextColumn**
  Returns the next column in display order. Column is the index of an item in the collection (a column).

- **GetNextVisibleColumn**
  Returns the next visible column in display order, Column is an index into the columns list.

- **GetOwner**
  Not documented.

- **GetPreviousColumn**
  Returns the previous column in display order, Column is an index into the columns list.

- **GetPreviousVisibleColumn**
  Returns the previous column in display order, Column is an index into the columns list.

- **GetVisibleColumns**
  Returns a list of all currently visible columns in actual order.

- **GetVisibleFixedWidth**
  Not documented.

- **HandleClick**
  Generates a click event if the mouse button has been released over the same column it was pressed first.

- **IndexChanged**
Called by a column when its index in the collection changes.

- **InitializePositionArray**
  Ensures that the column position array contains as much entries as columns are defined.

- **IsValidColumn**
  Determines whether the given column is valid or not, that is, whether it is one of the current columns.

- **LoadFromStream**
  Not documented.

- **PaintHeader**
  Not documented.

- **SaveToStream**
  Not documented.

- **TotalWidth**
  Not documented.

- **Update**
  Not documented.

- **UpdatePositions**
  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.

**Legend**

- `public`
- `Property`
- `read only`
- `protected`
- `Method`
- `virtual`

**Class Hierarchy**
File
VirtualTrees

Links
Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TVirtualTreeColumns.ClickIndex Property

Not documented.

Pascal

```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

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumns.Header Property**

**TVirtualTreeColumns Class**

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

**Links**

TVirtualTreeColumns Class

---

*What do you think about this topic? Send feedback!*
## TVirtualTreeColumns.HeaderBitmap Property

**TVirtualTreeColumns Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumns Class
**TVirtualTreeColumns.Items Property**

**TVirtualTreeColumns Class**

Not documented.

**Pascal**

```pascal
property Items [Index: TColumnIndex]: TVirtualTreeColumns;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualTreeColumns Class

**Links**

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumns.PositionToIndex**

**Property**

**TVirtualTreeColumns Class**

Not documented.

**Pascal**

```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**

**Links**

**TVirtualTreeColumns Class**

---

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumns::TrackIndex Property**

**TVirtualTreeColumns Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumns Class

---

*What do you think about this topic? [Send feedback]*
TVirtualTreeColumns.Add Method

TVirtualTreeColumns Class

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

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumns.AdjustAutoSize Method**

**TVirtualTreeColumns Class**

Called when columns must be sized so that the fit the client area.

**Pascal**

```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`) describes which column has just been resized.

**Class**

TVirtualTreeColumns Class

**Links**

TVirtualTreeColumns Class
**TVirtualTreeColumns.AdjustDownColumn Method**

**TVirtualTreeColumns Class**

Determines the column from the given position and returns it.

**Pascal**

```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**

**Links**

**TVirtualTreeColumns Class**

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumns.AdjustHoverColumn Method**

**TVirtualTreeColumns Class**

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

**Links**

TVirtualTreeColumns Class
**TVirtualTreeColumns.AdjustPosition Method**

**TVirtualTreeColumns Class**

Reorders the column position array so that the given column gets the given position.

**Pascal**

```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

**Links**

TVirtualTreeColumns Class

---

*What do you think about this topic? Send feedback!*
TVirtualTreeColumns.AnimatedResize Method

TVirtualTreeColumns Class

Resizes the given column animated by scrolling the window DC.

Pascal

```pascal
procedure AnimatedResize(Column: TColumnIndex; NewWidth: Integer);
```

Description

Resizes the given column animated by scrolling the window DC.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class
TVirtualTreeColumns.Assign Method

TVirtualTreeColumns Class

Not documented.

Pascal

```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

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.Clear Method

TVirtualTreeColumns Class

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

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.ColumnFromPosition Method (TColumnPosition)

TVirtualTreeColumns Class

Returns the index of the column at the given position.

Pascal

function ColumnFromPosition(PositionIndex: TColumnPosition): Integer;

Description

Returns the index of the column at the given position.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.ColumnFromPosition Method (TPoint, Boolean)

TVirtualTreeColumns Class

Determines the current column based on the position passed in P.

Pascal

```
function ColumnFromPosition(P: TPoint; Relative: Boolean): Integer;
```

Description

Determines the current column based on the position passed in P.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.Create Constructor

TVirtualTreeColumns Class

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

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.Destroy Destructor

TVirtualTreeColumns Class

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

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.DrawButtonText Method

TVirtualTreeColumns Class

Not documented.

Pascal

procedure DrawButtonText(DC: HDC; Caption: WideString);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.DrawXPButton Method

TVirtualTreeColumns Class

Helper procedure to draw an Windows XP like header button.

Pascal

```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

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumns.Equals Method**

**TVirtualTreeColumns Class**

Compares itself with the given set of columns.

**Pascal**

```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`

**Links**

`TVirtualTreeColumns Class`

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumns.FixPositions Method**

**TVirtualTreeColumns Class**

Fixes column positions after loading from DFM.

**Pascal**

```pascal
procedure FixPositions;
```

**Description**

Fixes column positions after loading from DFM.

**Class**

TVirtualTreeColumns Class

**Links**

TVirtualTreeColumns Class

*What do you think about this topic? Send feedback!*
TVirtualTreeColumns.GetColumnAndBounds Method

TVirtualTreeColumns Class

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, ColumnRight: Integer): Integer;
```

Description

*Left* and *Right* are undetermined if no column is involved.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.GetColumnBounds Method

TVirtualTreeColumns Class

Returns the left and right bound of the given column.

Pascal

```
procedure GetColumnBounds(Column: TColumnIndex; var
```

Description

If Column is NoColumn then the entire client width is returned.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.GetFirstVisibleColumn Method

TVirtualTreeColumns Class

Returns the index of the first visible column or "InvalidColumn" 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" if either no columns are defined or all columns are hidden.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class
TVirtualTreeColumns.GetLastVisibleColumn Method

TVirtualTreeColumns Class

Returns the index of the last visible column or "InvalidColumn" 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" if either no columns are defined or all columns are hidden.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumns.GetNextColumn Method**

*TVirtualTreeColumns Class*

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*

**Links**

*TVirtualTreeColumns Class*
TVirtualTreeColumns.GetNextVisibleColumn Method

TVirtualTreeColumns Class

Returns the next visible column in display order, Column is an index into the columns list.

Pascal

```pascal
function GetNextVisibleColumn(Column: TColumnIndex):
```

Description

Returns the next visible column in display order, Column is an index into the columns list.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.GetOwner Method

TVirtualTreeColumns Class

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

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumns.GetPreviousColumn Method**

**TVirtualTreeColumns Class**

Returns the previous column in display order, Column is an index into the columns list.

**Pascal**

```pascal
function GetPreviousColumn(Column: TColumnInfo): TColumnInfo;
```

**Description**

Returns the previous column in display order, Column is an index into the columns list.

**Class**

TVirtualTreeColumns Class

**Links**

TVirtualTreeColumns Class

*What do you think about this topic? Send feedback!*
TVirtualTreeColumns.GetPreviousVisibleColumn Method

TVirtualTreeColumns Class

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

Links

TVirtualTreeColumns Class
**TVirtualTreeColumns.GetVisibleColumns Method**

**TVirtualTreeColumns Class**

Returns a list of all currently visible columns in actual order.

**Pascal**

```pascal
function GetVisibleColumns: TColumnsArray;
```

**Description**

Returns a list of all currently visible columns in actual order.

**Class**

TVirtualTreeColumns Class

**Links**

TVirtualTreeColumns Class

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumns.GetVisibleFixedWidth Method**

*TVirtualTreeColumns Class*

Not documented.

**Pascal**

```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*

**Links**

*TVirtualTreeColumns Class*

---

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumns.HandleClick Method**

Generates a click event if the mouse button has been released over the same column it was pressed first.

**Pascal**

```pascal
procedure HandleClick(P: TPoint; Button: TMouseButton);
```

**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`
TVirtualTreeColumns.IndexChanged Method

TVirtualTreeColumns Class

Called by a column when its index in the collection changes.

Pascal

```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 the position array to reflect the change.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
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

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumns.IsValidColumn Method**

Type belongs to the `TVirtualTreeColumns` class.

Determine whether the given column is valid or not, that is, whether it is one of the current columns.

**Pascal**

```pascal
function IsValidColumn(Column: TColumnIndex): Boolean;
```

**Description**

Determine whether the given column is valid or not, that is, whether it is one of the current columns.

**Class**

- `TVirtualTreeColumns` class

**Links**

- `TVirtualTreeColumns` class

---

What do you think about this topic? Send feedback!
**TVirtualTreeColumns.LoadFromStream Method**

**TVirtualTreeColumns Class**

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumns Class

*What do you think about this topic? Send feedback!*
TVirtualTreeColumns.PaintHeader Method

TVirtualTreeColumns Class

Not documented.

Pascal

```
procedure PaintHeader(DC: HDC; R: TRect; HOffset: Integer);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
TVirtualTreeColumns.SaveToStream Method
TVirtualTreeColumns Class

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

Links
TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumns.TotalWidth Method**

**TVirtualTreeColumns Class**

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

**Links**

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
**TVirtualTreeColumns.Update Method**

**Class**

TVirtualTreeColumns Class

Not documented.

**Pascal**

```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

**Links**

TVirtualTreeColumns Class

---

What do you think about this topic? Send feedback!
TVirtualTreeColumns.UpdatePositions Method

TVirtualTreeColumns Class

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

Links

TVirtualTreeColumns Class

What do you think about this topic? Send feedback!
**TVirtualTreeHintWindow Class**

Internally used hint window class to support Unicode hints.

**Pascal**

```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 as 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

**Members**

**Methods**

- ![ActivateHint](https://example.com/green.png) **ActivateHint**
  
  Not documented.

- ![CalcHintRect](https://example.com/green.png) **CalcHintRect**
  
  Not documented.

- ![Create](https://example.com/green.png) **Create**
  
  Not documented.

- ![CreateParams](https://example.com/green.png) **CreateParams**
  
  Not documented.

- ![Destroy](https://example.com/green.png) **Destroy**
IsHintMsg
The VCL is a bit too generous when telling that an existing hint can be cancelled.

Paint
Not documented.

Legend

public
Method
virtual
protected

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Legend

What do you think about this topic? Send feedback!
**TVirtualTreeHintWindow.ActivateHint**

**Method**

TVirtualTreeHintWindow Class

Not documented.

**Pascal**

```
procedure ActivateHint(Rect: TRect; const AHint: string);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVirtualTreeHintWindow Class

**Links**

TVirtualTreeHintWindow Class

What do you think about this topic? Send feedback!
TVirtualTreeHintWindow.CalcHintRect Method

TVirtualTreeHintWindow Class

Not documented.

Pascal

```pascal
function CalcHintRect(MaxWidth: Integer; const AHint
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVirtualTreeHintWindow Class

Links

TVirtualTreeHintWindow Class

What do you think about this topic? Send feedback!
**TVirtualTreeHintWindow.Create Constructor**

**TVirtualTreeHintWindow Class**

Not documented.

**Pascal**

```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**

**Links**

**TVirtualTreeHintWindow Class**

*What do you think about this topic? Send feedback!*
**TVirtualTreeHintWindow.CreateParams Method**

**TVirtualTreeHintWindow Class**

Not documented.

**Pascal**

```pascal
procedure CreateParams(var Params: TCreateParams);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

**TVirtualTreeHintWindow Class**

**Links**

**TVirtualTreeHintWindow Class**

*What do you think about this topic? Send feedback!*
**TVirtualTreeHintWindow.Destroy**

**Destructor**

**TVirtualTreeHintWindow Class**

Not documented.

**Pascal**

```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**

**Links**

**TVirtualTreeHintWindow Class**

What do you think about this topic? Send feedback!
The VCL is a bit too generous when telling that an existing hint can be cancelled.

Pascal

```pascal
function IsHintMsg(var Msg: TMsg): Boolean; override
```

Description
Need to specify further here.

Class
TVirtualTreeHintWindow Class

Links
TVirtualTreeHintWindow Class

What do you think about this topic? Send feedback!
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

Links

TVirtualTreeHintWindow Class

What do you think about this topic? Send feedback!
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 but is the publicly available class.

Group

Classes

Members

Properties

- **AnimationOptions**
  Options related to animations.
- **AutoOptions**
  Options related to automatic actions.
- **MiscOptions**
  Options not related to any other category.
- **PaintOptions**
  Options related to painting.
- **SelectionOptions**
  Options related to the way nodes can be selected.
**TCustomVirtualTreeOptions Class**

- **AnimationOptions**
  Options related to animations.
- **AutoOptions**
  Options related to automatic actions.
- **MiscOptions**
  Options not related to any other category.
- **Owner**
  Owner tree to which the property class belongs.
- **PaintOptions**
  Options related to painting.
- **SelectionOptions**
  Options related to the way nodes can be selected.

**Methods**

**TCustomVirtualTreeOptions Class**

- **AssignTo**
  Used to copy this option class to another option collection.
- **Create**
  Constructor of the class.

**Legend**

- published
- Property
- protected
- public
- read only
- Method
virtual

Class Hierarchy

File

VirtualTrees

Links

Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TVirtualTreeOptions.AnimationOptions Property

TVirtualTreeOptions Class

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

Links

TVirtualTreeOptions Class
**TVirtualTreeOptions.AutoOptions Property**

`TVirtualTreeOptions.Class` Options related to automatic actions.

**Pascal**

```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`

**Links**

`TVirtualTreeOptions.Class`

---

*What do you think about this topic? Send feedback!*
TVirtualTreeOptions.MiscOptions Property

Options not related to any other category.

Pascal

```pascal
property MiscOptions: TVTMiscOptions;
```

Description

These options can be used to switch miscellaneous aspects in a tree.

Class

TVirtualTreeOptions Class

Links

TVirtualTreeOptions Class

What do you think about this topic? Send feedback!
**TVirtualTreeOptions.PaintOptions Property**

**TVirtualTreeOptions Class**

Options related to painting.

**Pascal**

```pascal
property PaintOptions: TVTPaintOptions;
```

**Description**

These options can be used to switch visual aspects of a tree.

**Class**

TVirtualTreeOptions Class

**Links**

TVirtualTreeOptions Class

*What do you think about this topic? Send feedback!*
TVirtualTreeOptions.SelectionOptions Property

TVirtualTreeOptions Class

Options related to the way nodes can be selected.

Pascal

```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

Links

TVirtualTreeOptions Class

What do you think about this topic? Send feedback!
TVTColors Class

Collects all color related options for the tree control.

Pascal

```pascal
TVTColors = class(TPersistent);
```

Description

TVTColors makes it much more convenient to adjust Virtual Treeview's colors. Since everything is in one place you can also easily compare all colors.

Group

Classes

Members

Properties

- **BorderColor**
  Not documented.
- **DisabledColor**
  Not documented.
- **DropMarkColor**
  Color of the drop mark.
- **DropTargetBorderColor**
  Not documented.
- **DropTargetColor**
  Not documented.
- **FocusedSelectionBorderColor**
Not documented.

- **FocusedSelectionColor**
  Not documented.

- **GridLineColor**
  Not documented.

- **HeaderHotColor**
  Not documented.

- **HotColor**
  Not documented.

- **SelectionRectangleBlendColor**
  Not documented.

- **SelectionRectangleBorderColor**
  Not documented.

- **TreeLineColor**
  Not documented.

- **UnfocusedSelectionBorderColor**
  Not documented.

- **UnfocusedSelectionColor**
  Not documented.

**Methods**

- **Assign**
  Not documented.

- **Create**
  Not documented.

**Legend**

- published
- Property
- public
- Method
virtual

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TVTColors.BorderColor Property

TVTColors Class

Not documented.

Pascal

```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

Links

TVTColors Class

What do you think about this topic? Send feedback!
**TVTColors.DisabledColor Property**

**TVTColors Class**

Not documented.

**Pascal**

```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

**Links**

TVTColors Class

What do you think about this topic? Send feedback!
**TVTCols.DropMarkColor Property**

**TVTCols Class**

Color of the drop mark.

**Pascal**

```pascal
property DropMarkColor: TColor;
```

**Description**

Since the drop metaphor 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**

TVTCols Class

**Links**

TVTCols Class

---

What do you think about this topic? Send feedback!
**TVTColors.DropTargetBorderColor**

**Property**

**TVTColors Class**

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`

**Links**

`TVTColors Class`

---

*What do you think about this topic? [Send feedback]*
**TVTColors.DropTargetColor Property**

**TVTColors Class**

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

**Links**

TVTColors Class

---

*What do you think about this topic? Send feedback!*
TVTColors.FocusedSelectionBorderColor Property

TVTColors Class

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

Links

    TVTColors Class

What do you think about this topic? Send feedback!
TVTColors.FocusedSelectionColor

Property

TVTColors Class

Not documented.

Pascal

```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

Links

TVTColors Class

What do you think about this topic? Send feedback!
TVTColors.GridLineColor Property

TVTColors Class

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

Links

TVTColors Class

What do you think about this topic? Send feedback!
**TVTColors.HeaderHotColor Property**

**TVTColors Class**

Not documented.

**Pascal**

```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

**Links**

TVTColors Class

*What do you think about this topic? Send feedback!*
TVTColors.HotColor Property

TVTColors Class

Not documented.

Pascal

```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

Links

TVTColors Class

What do you think about this topic? Send feedback!
**TVTColors.SelectionRectangleBlendColor**

**Property**

**TVTColors Class**

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

**Links**

TVTColors Class

*What do you think about this topic? Send feedback!*
TVTColors.SelectionRectangleBorderColor Property

TVTColors Class

Not documented.

Pascal

```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

Links

TVTColors Class

What do you think about this topic? Send feedback!
TVTColors.TreeLineColor Property

TVTColors Class

Not documented.

Pascal

```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

Links

TVTColors Class

What do you think about this topic? Send feedback!
TVTColors.

**UnfocusedSelectionBorderColor Property**

TVTColors Class

Not documented.

**Pascal**

```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

**Links**

TVTColors Class

---

*What do you think about this topic? Send feedback!*
**TVTColors.UnfocusedSelectionColorProperty**

**TVTColors Class**

Not documented.

**Pascal**

```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

**Links**

TVTColors Class

*What do you think about this topic? Send feedback!*
TVTColors.Assign Method

TVTColors Class

Not documented.

Pascal

```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

Links

TVTColors Class

What do you think about this topic? Send feedback!
TVTColors.Create Constructor

**TVTColors Class**

Not documented.

**Pascal**

```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

**Links**

TVTColors Class

*What do you think about this topic? Send feedback!*
TVTDataObject Class

Implementation of an IDataObject interface.

Pascal

```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

**Members**

**Properties**

- ![Not documented] ForClipboard
  - Not documented.
- ![Not documented] FormatEtcArray
  - Not documented.
- ![Not documented] InternalStgMediumArray
  - Not documented.
- ![Not documented] Owner
  - Not documented.
Methods

- **CanonicalIUnknown**
  Helper method for setting data in the IDataObject.
- **Create**
  Constructor of the class.
- **DAdvise**
  Implementation of the IDataObject.DAdvise method.
- **Destroy**
  Destructor of the class.
- **DUnadvise**
  Implementation of the IDataObject.DUnAdvise method.
- **EnumDAdvise**
  Implementation of the IDataObject.EnumDAdvise method.
- **EnumFormatEtc**
  Implementation of the IDataObject.EnumFormatEtc method.
- **EqualFormatEtc**
  Compares two TFormatEtc structures.
- **FindFormatEtc**
  Searches the given array for a the given format.
- **FindInternalStgMedium**
  Returns a storage medium for a given clipboard format.
- **GetCanonicalFormatEtc**
  Implementation of the IDataObject.GetCanonicalFormatEtc method.
- **GetData**
  Implementation of the IDataObject.GetData method.
- **GetDataHere**
  Implementation of the IDataObject.GetDataHere method.
- **HGlobalClone**
  Helper method for **SetData**.
- **QueryGetData**
  Implementation of the IDataObject.QueryGetData method.
- **RenderInternalOLEData**
  Helper method to return data previously stored by **SetData**.
- **SetData**
Implementation of the IDataObject.SetData method.

StgMediumIncRef
Central managing method to copy OLE data.

Legend

- protected
- Property
- read only
- Method
- public
- virtual

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TVTDataObject.ForClipboard Property

TVTDataObject Class

Not documented.

Pascal

```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

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.FormatEtcArray Property

TVTDataObject Class

Not documented.

Pascal

```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

Links

TVTDataObject Class

*What do you think about this topic? Send feedback!*
TVTDataObject.InternalStgMediumArray Property

TVTDataObject Class

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

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.Owner Property

TVTDataObject Class

Not documented.

Pascal

```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

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
**TVTDataObject.CanonicalIUnknown Method**

**TVTDataObject Class**

Helper method for setting data in the IDataObject.

**Pascal**

```pascal
function CanonicalIUnknown(TestUnknown: IUnknown): IUnknown;
```

**Description**

In **SetData** the class can get a circular reference if the client calls **GetData** then calls **SetData** with the same StgMedium. Because the unkForRelease for the IDataObject can be marshalled it is necessary to get pointers that can be correctly compared. CanonicalIUnknown 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** article by Raymond Chen at MSDN.

**Class**

TVTDataObject Class

**Links**

TVTDataObject Class
TVTDataObject.Create Constructor

**TVTDataObject Class**

Constructor of the class.

**Pascal**

```
constructor Create(AOwner: TBaseVirtualTree; ForClip)
```

**Description**

Create is used only for initialization.

**Class**

TVTDataObject Class

**Links**

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.DAdvise Method

TVTDataObject Class

Implementation of the IDataObject.DAdvise method.

Pascal

function DAdvise(const FormatEtc: TFormatEtc; advf: Integer): Integer;

Description

Advise sinks are used to have an opportunity for clients to get notified if something changes in the data object. TVTDataObject uses the data advise holder APIs to provide the advise sink service.

Class

TVTDataObject Class

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.Destroy Destructor

**TVTDataObject Class**

Destructor of the class.

**Pascal**

```
destructor Destroy; override;
```

**Description**

Cleans up the object.

**Class**

TVTDataObject Class

**Links**

TVTDataObject Class

What do you think about this topic? Send feedback!
**TVTDataObject.DUnadvise Method**

**TVTDataObject Class**

Implementation of the IDataObject.DUnAdvise method.

**Pascal**

```pascal
function DUnadvise(dwConnection: Integer): HResult;
```

**Description**

DUnadvise reverses the call to DAdvise.

**Class**

TVTDataObject Class

**Links**

TVTDataObject Class

---

*What do you think about this topic? Send feedback!*
TVTDDataObject.EnumDAdvise Method

TVTDDataObject Class

Implementation of the IDataObject.EnumDAdvise method.

Pascal

```pascal
function EnumDAdvise(out enumAdvise: IEnumStatData): HResult;
```

Description

EnumDAdvise 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

TVTDDataObject Class

Links

TVTDDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.EnumFormatEtc Method

TVTDataObject Class

Implementation of the IDataObject.EnumFormatEtc method.

Pascal

```pascal
function EnumFormatEtc(Direction: Integer; out EnumFormatEtc: IEnumFormatEtc): HResult;
```

Description

This method creates a FormatEtc enumerator class which is used to enumerate all data formats supported by the owner tree.

Class

TVTDataObject Class

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
**TVTDataObject.EqualFormatEtc Method**

TVTDataObject Class

Compares two TFormatEtc structures.

**Pascal**

```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

**Links**

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.FindFormatEtc Method

TVTDataObject Class

Searchs the given array for a the given format.

Pascal

```pascal
function FindFormatEtc(TestFormatEtc: TFormaetEtc; const Description: string): boolean;
```

Description

Returns true if the given format is part of the array.

Class

TVTDataObject Class

Links

TVTDataObject Class
TVTDataObject.FindInternalStgMedium Method

TVTDataObject Class

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

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.GetCanonicalFormatEtc Method

Implementation of the IDataObject.GetCanonicalFormatEtc method.

Pascal

```pascal
function GetCanonicalFormatEtc(const FormatEtc: TFormatEtc): TFormatEtc;
```

Description

The implementation of this method simply consists of a result value telling the caller to use the EnumFormatEtc method.

Class

TVTDataObject Class

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.GetData Method

TVTDataObject Class | See Also

Implementation of the IDataObject.GetData method.

Pascal

function GetData(const FormatEtcIn: TFormatEtc; out Medium): HResult;

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 format and any data previously set by the SetData 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

Links
TVTDataObject Class, See Also

What do you think about this topic? Send feedback!
TVTDataObject.GetDataHere Method

TVTDataObject Class

Implementation of the IDataObject.GetDataHere method.

Pascal

```
function GetDataHere(const FormatEtc: TFormatEtc; out Description: TDESC; out DescriptionEx: TDESCRIPTIONINFO): BOOL;
```

Description

GetDataHere is an alternative data retrieval method to GetData, 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 implemented.

Class

TVTDataObject Class

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.HGlobalClone Method

TVTDataObject Class

Helper method for SetData.

Pascal

function HGlobalClone(HGlobal: THandle): THandle;

Description

This method copies a HGlobal memory block.

Class

TVTDataObject Class

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.QueryGetData Method

Implementation of the IDataObject.QueryGetData method.

Pascal

```
function QueryGetData(const FormatEtc: TFormatEtc): HRESULT;
```

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**

- Virtual Treeview reference and process identifier
- native serialized tree data

**TCustomVirtualStringTree**
• 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

Links
TVTDataObject Class

What do you think about this topic? Send feedback!
**TVTDataObject.RenderInternalOLEData Method**

Helper method to return data previously stored by **SetData**.

**Pascal**

```pascal
function RenderInternalOLEData(const FormatEtcIn: TFormatEtc): TVTDataObject;
```

**Description**

For some operations (e.g. shell transfers with **IDropTargetHelper** interface) data is stored in the class. **RenderInternalOLEData** returns this data when queried later.

**Class**

TVTDataObject Class

**Links**

TVTDataObject Class

---

*What do you think about this topic? Send feedback!*
TVTDataObject.SetData Method

TVTDataObject Class

Implementation of the IDataObject.SetData method.

Pascal

function SetData(const FormatEtc: TFormatEtc; var Medium: TStgMedium; DoRelease: BOOL): HRESULT;

Description

This method is used to add or replace data in the data object.

Class

TVTDataObject Class

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDataObject.StgMediumIncRef Method

TVTDataObject Class

Central managing method to copy OLE data.

Pascal

```pascal
function StgMediumIncRef(const InStgMedium: TStgMedium): integer;
```

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

Links

TVTDataObject Class

What do you think about this topic? Send feedback!
TVTDragImage Class

Classes | Methods | Properties | Legend

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

Members

Properties

- ColorKey
  - Not documented.
- Fade
  - Not documented.
- MoveRestriction
  - Not documented.
- PostBlendBias
  - Not documented.
- PreBlendBias
  - Not documented.
- Transparency
  - Not documented.
Visible
Not documented.

Methods

Create
Not documented.
Destroy
Not documented.
DragTo
Moves the drag image to a new position, which is determined from the passed point $P$ and the previous mouse position.
EndDrag
Not documented.
GetDragImageRect
Returns the current size and position of the drag image (screen coordinates).
HideDragImage
Not documented.
InternalShowDragImage
Frequently called helper routine to actually do the blend and put it onto
MakeAlphaChannel
Not documented.
PrepareDrag
Creates all necessary structures to do alpha blended dragging using the given image.
RecaptureBackground
Notification by the drop target tree to update the background image because something in the tree has changed.
ShowDragImage
Shows the drag image after it has been hidden by HideDragImage.
WillMove
Add a summary here...

Legend
public
Property
read only
Method
virtual
protected

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TVTDragImage.ColorKey Property

TVTDragImage Class

Not documented.

Pascal

```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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.Fade Property

TVTDragImage Class

Not documented.

Pascal

```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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.MoveRestriction Property

TVTDragImage Class

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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.PostBlendBias Property

TVTDragImage Class

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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.PreBlendBias Property
TVTDragImage Class

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

Links
TVTDragImage Class
TVTDragImage.Transparency Property

TVTDragImage Class

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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.Visible Property

TVTDragImage Class

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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.Create Constructor
TVTDragImage Class

Not documented.

Pascal

```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

Links
TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.Destroy Destructor

TVTDragImage Class

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

Links
TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.DragTo Method
TVTDragImage Class

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

Links

TVTDragImage Class

*What do you think about this topic? Send feedback!*
TVTDragImage.EndDrag Method

TVTDragImage Class

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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
**TVTDragImage.GetDragImageRect**

TVTDragImage Class

Returns the current size and position of the drag image (screen coordinates).

**Pascal**

```pascal
function GetDragImageRect: TRect;
```

**Description**

Returns the current size and position of the drag image (screen coordinates).

**Class**

TVTDragImage Class

**Links**

TVTDragImage Class

*What do you think about this topic? Send feedback!*
TVTDragImage.HideDragImage Method

TVTDragImage Class

Not documented.

Pascal

```pascal
procedure HideDragImage;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragImage Class

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.InternalShowDragImage Method

TVTDragImage Class

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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.MakeAlphaChannel Method

TVTDragImage Class

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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.PrepareDrag Method

TVTDragImage Class

Creates all necessary structures to do alpha blended dragging using the given image.

Pascal

```
procedure PrepareDrag(DragImage: TBitmap; ImagePosition: TPoint; HotSpot: TPoint;
```

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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.RecaptureBackground Method

TVTDragImage Class

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

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.ShowDragImage Method

TVTDragImage Class

Shows the drag image after it has been hidden by HideDragImage.

Pascal

procedure ShowDragImage;

Description

Also this method does nothing if the system manages the drag image.

Class

TVTDragImage Class

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragImage.WillMove Method

TVTDragImage Class

Add a summary here...

Pascal

```pascal
function WillMove(P: TPoint): Boolean;
```

Description

This method determines whether the drag image would "physically" move when DragTo would be called with the same target point. Always returns false if the system drag image support is available.

Class

TVTDragImage Class

Links

TVTDragImage Class

What do you think about this topic? Send feedback!
TVTDragManager Class

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Pascal

```pascal
TVTDragManager = class(TInterfacedObject, IVTDragManager)
```

Members

Methods
- ![Create](#) Not documented.
- ![Destroy](#) Not documented.
- ![DragEnter](#) Not documented.
- ![DragLeave](#) Not documented.
- ![DragOver](#) Not documented.
- ![Drop](#) Not documented.
ForceDragLeave
This method calls the drop target helper's DragLeave method to ensure it removes the drag image from screen.

GiveFeedback
Not documented.

QueryContinueDrag
Not documented.

IVTDragManager Interface

ForceDragLeave
Not documented.

GetDataObject
Not documented.

GetDragSource
Not documented.

GetDropTargetHelperSupported
Not documented.

GetIsDropTarget
Not documented.

Properties
IVTDragManager Interface

DataObject
Not documented.

DragSource
Not documented.

DropTargetHelperSupported
Not documented.

IsDropTarget
Not documented.

Legend
public
Method
virtual
Property
read only

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
**TVTDragManager.Create Constructor**

**TVTDragManager Class**

Not documented.

**Pascal**

```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

**Links**

TVTDragManager Class

---

*What do you think about this topic?*  
Send feedback!
TVTDragManager.Destory Destructor

TVTDragManager Class

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

Links

TVTDragManager Class

What do you think about this topic? Send feedback!
TVTDragManager.DragEnter Method

TVTDragManager Class

Not documented.

Pascal

function DragEnter(const DataObject: IDataObject; KeyState: Integer; Pt: TPoint);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class

Links

TVTDragManager Class

What do you think about this topic? Send feedback!
**TVTDragManager.DragLeave Method**

**TVTDragManager Class**

Not documented.

**Pascal**

```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

**Links**

TVTDragManager Class

*What do you think about this topic? Send feedback!*
TVTDragManager.DragOver Method

TVTDragManager Class

Not documented.

Pascal

```pascal
function DragOver(KeyState: Integer; Pt: TPoint; var Effect: Integer): HResult;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class

Links

TVTDragManager Class

What do you think about this topic? Send feedback!
TVTDragManager.Drop Method

TVTDragManager Class

Not documented.

Pascal

```
function Drop(const DataObject: IDataObject; KeyState: Integer; Pt: TPoint): Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class

Links

TVTDragManager Class

What do you think about this topic? Send feedback!
TVTDragManager.ForceDragLeave Method

TVTDragManager Class

This method calls the drop target helper's DragLeave method to ensure it removes the drag image from screen.

Pascal

procedure ForceDragLeave; stdcall;

Description

This method calls the drop target helper's DragLeave method to ensure it removes the drag image from screen.

Class

TVTDragManager Class

Links

TVTDragManager Class

What do you think about this topic? Send feedback!
**TVTDragManager.GiveFeedback Method**

**TVTDragManager Class**

Not documented.

**Pascal**

```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

**Links**

TVTDragManager Class

---

*What do you think about this topic? Send feedback!*
TVTDragManager.QueryContinueDrag Method
TVTDragManager Class

Not documented.

Pascal

```pascal
function QueryContinueDrag(EscapePressed: BOOL; KeyState: Integer): HRESULT;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTDragManager Class

Links

TVTDragManager Class

What do you think about this topic? Send feedback!
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

Members

Properties

- **AutoSelect**
  Not documented.
- **AutoSize**
  Not documented.
- **BorderStyle**
  Not documented.
- **CharCase**
  Not documented.
- **HideSelection**
  Not documented.
- **MaxLength**
  Not documented.
OEMConvert
Not documented.
PasswordChar
Not documented.

Methods

AutoAdjustSize
Not documented.
Create
Not documented.
CreateParams
Not documented.
Release
Not documented.

Legend

public
Property
protected
Method
virtual

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Properties, Legend
What do you think about this topic? Send feedback!
TVTEdit Class

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

Links

TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.AutoSize Property
TVTEdit Class

Not documented.

Pascal

```pascal
property AutoSize;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class

Links

TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.BorderStyle Property

TVTEdit Class

Not documented.

Pascal

```pascal
property BorderStyle;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class

Links

TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.CharCase Property
TVTEdit Class

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

Links
TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.HideSelection Property
TVTEdit Class

Not documented.

Pascal

```pascal
property HideSelection;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class

Links

TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.MaxLength Property

TVTEdit Class

Not documented.

Pascal

```pascal
property MaxLength;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class

Links

TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.OEMConvert Property
TVTEdit Class

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

Links

TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.PasswordChar Property

TVTEdit Class

Not documented.

Pascal

```pascal
property PasswordChar;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class

Links

TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.AutoAdjustSize Method

TVTEdit Class

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

Links

TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.Create Constructor

TVTEdit Class

Not documented.

Pascal

```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

Links

TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.CreateParams Method

TVTEdit Class

Not documented.

Pascal

```pascal
procedure CreateParams(var Params: TCreateParams);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTEdit Class

Links

TVTEdit Class

What do you think about this topic? Send feedback!
TVTEdit.Release Method

TVTEdit Class

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

Links

TVTEdit Class

What do you think about this topic? Send feedback!
**TVTHeader Class**

Classes | Methods | Properties | Legend

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

**Members**

**Properties**

- AutoSizeIndex
  - Not documented.
- Background
  - Not documented.
- Columns
  - Not documented.
- DragImage
  - Not documented.
- Font
  - Not documented.
- Height
  - Not documented.
Images
Not documented.

MainColumn
Not documented.

Options
Not documented.

ParentFont
Not documented.

PopupMenu
Not documented.

SortColumn
Not documented.

SortDirection
Not documented.

States
Not documented.

Style
Not documented.

Treeview
Not documented.

UseColumns
Not documented.

Methods

Assign
Not documented.

AutoFitColumns
Not documented.

CanWriteColumns
Not documented.

ChangeScale
Not documented.

Create
Not documented.
**Destroy**
Not documented.

**DetermineSplitterIndex**
Tries to find the index of that column whose right border corresponds to \( P \).

**DragTo**
Moves the drag image to a new position, which is determined from the passed point \( P \) and the previous mouse position.

**GetColumnsClass**
Returns the class to be used for the actual column implementation.

**GetOwner**
Not documented.

**GetShiftState**
Not documented.

**HandleHeaderMouseMove**
Not documented.

**HandleMessage**
General message handler for the header.

**ImageListChange**
Not documented.

**InHeader**
Determines whether the given point (client coordinates!) is within the header rectangle (non-client coordinates).

**Invalidate**
Invalidates the entire header or parts of it so they are repainted.

**LoadFromStream**
Restores the state of the header from the given stream.

**PrepareDrag**
Initializes dragging of the header, \( P \) is the current mouse postion and \textbf{Start} the initial mouse position.

**ReadColumns**
Not documented.

**RecalculateHeader**
Initiate a recalculation of the non-client area of the owner tree.

**RestoreColumns**
Restores all columns to their width which they had before they have been auto fitted.

- **SaveToStream**
  Saves the complete state of the header into the provided stream.

- **UpdateMainColumn**
  Called once the load process of the owner tree is done.

- **UpdateSpringColumns**
  Not documented.

- **WriteColumns**
  Not documented.

**Legend**

- published
- Property
- public
- read only
- Method
- virtual
- protected

**Class Hierarchy**

[Diagram: TPersistent ➔ TVTHeader]

**File**

- VirtualTrees

**Links**

- Classes, Methods, Properties, Legend

---

*What do you think about this topic? Send feedback!*
**TVTHeader.AutoScaleIndex Property**

**TVTHeader Class**

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

**Links**

TVTHeader Class

---

*What do you think about this topic? Send feedback!*
TVTHeader.Background Property

TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.Columns Property
TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHdr.DragImage Property

TVTHdr Class

Not documented.

Pascal

\[ \text{property } \text{DragImage: TVTDragImage;} \]

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHdr Class

Links

TVTHdr Class

What do you think about this topic? Send feedback!
**TVTHeader.Font Property**

TVTHeader Class

Not documented.

Pascal

```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

**Links**

TVTHeader Class

---

*What do you think about this topic? Send feedback!*
TVTHeader.Height Property
TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.Images Property

TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.MainColumn Property

TVTHeader Class

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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.Options Property

TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
**TVTHeader.PopupMenu Property**

*TVTHeader Class*

Not documented.

**Pascal**

```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*

**Links**

*TVTHeader Class*

---

What do you think about this topic? Send feedback!
TVTHeader.PopupMenu Property

TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic?  Send feedback!
TVTHeader.SortColumn Property

TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.SortDirection Property

TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
**TVTHeader.States Property**

**TVTHeader Class**

Not documented.

**Pascal**

```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

**Links**

TVTHeader Class

*What do you think about this topic? Send feedback!*
TVTHHeader.Style Property

TVTHHeader Class

Not documented.

Pascal

property Style: TVTHHeaderStyle;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHHeader Class

Links

TVTHHeader Class

What do you think about this topic? Send feedback!
**TVTHeader.Treeview Property**

**TVTHeader Class**

Not documented.

**Pascal**

```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

**Links**

TVTHeader Class

---

*What do you think about this topic? Send feedback!*
TVTHeader.UseColumns Property

TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.Assign Method

TVTHeader Class

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

Links
TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.AutoFitColumns Method

TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class
TVTHheader.CanWriteColumns Method

TVTHheader Class

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
TVTHheader Class

Links
TVTHheader Class

What do you think about this topic? Send feedback!
**TVTHeader.ChangeScale Method**

**TVTHeader Class**

Not documented.

**Pascal**

```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

**Links**

TVTHeader Class

*What do you think about this topic? Send feedback!*
**TVTHeader.Create Constructor**

**TVTHeader Class**

Not documented.

**Pascal**

```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

**Links**

TVTHeader Class

---

*What do you think about this topic? Send feedback!*
TVTHeader.Destroy Destructor

TVTHeader Class

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

Links
TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.DetermineSplitterIndex Method

TVTHeader Class

Tries to find the index of that column whose right border corresponds to P.

Pascal

```
function DetermineSplitterIndex(P: TPoint): Boolean;
```

Description

Result is true 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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
**TVTHeader.DragTo Method**

**TVTHeader Class**

Moves the drag image to a new position, which is determined from the passed point \( P \) and the previous mouse position.

**Pascal**

```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

**Links**

TVTHeader Class

---

What do you think about this topic? Send feedback!
**TVTHeader.GetColumnsClass Method**

**TVTHeader Class**

Returns the class to be used for the actual column implementation.

**Pascal**

```pascal
define GetColumnsClass: TVirtualTreeColumnsClass;
```

**Description**

Descendants may optionally override this and return their own class.

**Class**

TVTHeader Class

**Links**

TVTHeader Class

*What do you think about this topic? Send feedback!*
**TVTHeader.GetOwner Method**

**TVTHeader Class**

Not documented.

**Pascal**

```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

**Links**

TVTHeader Class

*What do you think about this topic? Send feedback!*
TVTHeader.GetShiftState Method

TVTHeader Class

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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.HandleHeaderMouseMove Method

TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.HandleMessage Method

TVTHeader Class

General message handler for the header.

Pascal

```
function HandleMessage(var Message: TMessage): Boolean;
```

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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.ImageListChange Method

TVTHeader Class

Not documented.

Pascal

```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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.InHeader Method

TVTHeader Class

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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.Invalidate Method

TVTHeader Class

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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
**TVTHeader.LoadFromStream Method**  
*TVTHeader Class*

Restores the state of the header from the given stream.

**Pascal**

```pascal
procedure LoadFromStream(const Stream: TStream); virtual
```

**Description**

Restores the state of the header from the given stream.

**Class**

*TVTHeader Class*

**Links**

*TVTHeader Class*

*What do you think about this topic? Send feedback!*
TVTHeader.PrepareDrag Method

TVTHeader Class

Initializes dragging of the header, \( P \) is the current mouse postion and \textit{Start} the initial mouse position.

Pascal

\begin{verbatim}
procedure PrepareDrag(P: TPoint; Start: TPoint);
\end{verbatim}

Description

Initializes dragging of the header, \( P \) is the current mouse postion and \textit{Start} the initial mouse position.

Class

TVTHeader Class

Links

TVTHeader Class

\textit{What do you think about this topic? Send feedback!}
TVTHeader.ReadColumns Method

TVTHeader Class

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

Links

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.RecalculateHeader Method

TVTHeader Class

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

Links

TVTHeader Class
**TVTHeader.RestoreColumns Method**

**TVTHeader Class**

Restores all columns to their width which they had before they have been auto fitted.

**Pascal**

```pascal
procedure RestoreColumns;
```

**Description**

Restores all columns to their width which they had before they have been auto fitted.

**Class**

TVTHeader Class

**Links**

TVTHeader Class

---

*What do you think about this topic? Send feedback!*
TVTHeader.SaveToStream Method

TVTHeader Class

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

**Links**

TVTHeader Class

*What do you think about this topic? Send feedback!*
**TVTHeader.UpdateMainColumn Method**

**TVTHeader Class**

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

**Links**

TVTHeader Class

What do you think about this topic? Send feedback!
TVTHeader.UpdateSpringColumns Method
TVTHeader Class

Not documented.

Pascal

```pascal
procedure UpdateSpringColumns;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TVTHeader Class

Links

TVTHeader Class

What do you think about this topic? Send feedback!
**TVTHeader.WriteColumns Method**

**TVTHeader Class**

Not documented.

**Pascal**

```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

**Links**

TVTHeader Class

*What do you think about this topic? Send feedback!*
TVTHeaderPopupMenu Class

Events | Classes | Methods | Properties | Legend

Not documented.

Pascal

```pascal
TVTHeaderPopupMenu = class(TPopupMenu);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes

Members

Events

- ![Event](image)
  - **OnAddHeaderPopupItem**
    - Not documented.
- ![Event](image)
  - **OnColumnChange**
    - Not documented.

Properties

- ![Property](image)
  - **Options**
    - Not documented.

Methods

- ![Method](image)
  - **DoAddHeaderPopupItem**
Not documented.

- **DoColumnChange**
  - Not documented.
- **OnMenuItemClick**
  - Not documented.
- **Popup**
  - Not documented.

**Legend**

- published
- Event
- Property
- protected
- Method
- virtual
- public

**Class Hierarchy**

[VTHeaderPopup](#)

**File**

VTHeaderPopup

**Links**

- [Events](#), [Classes](#), [Methods](#), [Properties](#), [Legend](#)

---

*What do you think about this topic?* Send feedback!
TVTHeaderPopupMenu.OnAddHeaderPopupItem Event
TVTHeaderPopupMenu Class

Not documented.

Pascal

```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

Links

TVTHeaderPopupMenu Class

What do you think about this topic? Send feedback!
TVTHeaderPopupMenu class

TVTHeaderPopupMenu.OnColumnChange Event

TVTHeaderPopupMenu Class

Not documented.

Pascal

```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

Links

TVTHeaderPopupMenu Class

What do you think about this topic? Send feedback!
TVTHeaderPopupMenu.Options Property

Not documented.

Pascal

```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

Links

TVTHeaderPopupMenu Class

What do you think about this topic? Send feedback!
TVTHeaderPopupMenu.DoAddHeaderPopupItem

**TVTHeaderPopupMenu Class**

Not documented.

**Pascal**

```pascal
procedure DoAddHeaderPopupItem(const Column: TColumn);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVTHeaderPopupMenu Class

**Links**

TVTHeaderPopupMenu Class

---

*What do you think about this topic? Send feedback!*
**TVTHeaderPopupMenu.DoColumnChange Method**

**TVTHeaderPopupMenu Class**

Not documented.

**Pascal**

```pascal
procedure DoColumnChange(Column: TColumnInfo; Visible: Boolean);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Class**

TVTHeaderPopupMenu Class

**Links**

TVTHeaderPopupMenu Class

*What do you think about this topic? Send feedback!*
TVTHeaderPopupMenu.OnMenuItemClick Method

TVTHeaderPopupMenu Class

Not documented.

Pascal

```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

Links

TVTHeaderPopupMenu Class

What do you think about this topic? Send feedback!
TVTHeaderPopupMenu.Popup Method

Not documented.

Pascal

```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

Links

TVTHeaderPopupMenu Class

What do you think about this topic? Send feedback!
TWideBufferedString Class

Pascal

TWideBufferedString = class;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Classes

Members

Properties

AsString

Not documented.

Methods

Add

Not documented.

AddNewLine

Not documented.

Destroy

Not documented.

Legend
public

Property
read only

Method
virtual

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
**TWideBufferedString.AsString Property**

**TWideBufferedString Class**

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

**Links**

TWideBufferedString Class

*What do you think about this topic?* Send feedback!
TWideBufferedString.Add Method

TWideBufferedString Class

Not documented.

Pascal

```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

Links

TWideBufferedString Class

What do you think about this topic? Send feedback!
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

Links

TWideBufferedString Class

What do you think about this topic? Send feedback!
**TWideBufferedString.Destroy Destructor**

**TWideBufferedString Class**

Not documented.

**Pascal**

```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**

**Links**

**TWideBufferedString Class**

What do you think about this topic? Send feedback!
TWorkerThread Class

Classes | Methods | Properties | Legend

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

Members

Properties

- CurrentTree
  Not documented.

Methods

- AddTree
  Not documented.
- ChangeTreeStates
  Not documented.
- Create
  Not documented.
- Destroy
  Not documented.
Execute
Not documented.

RemoveTree
Not documented.

Legend

- public
- Property
- read only
- Method
- protected
- virtual

Class Hierarchy

File
VirtualTrees

Links
Classes, Methods, Properties, Legend

What do you think about this topic? Send feedback!
TWorkerThread.CurrentTree Property
TWorkerThread Class

Not documented.

Pascal

```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

Links
TWorkerThread Class

What do you think about this topic? Send feedback!
TWorkerThread.AddTree Method

TWorkerThread Class

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

Links

TWorkerThread Class

What do you think about this topic? Send feedback!
TWorkerThread.ChangeTreeStates Method

TWorkerThread Class

Not documented.

Pascal

```
procedure ChangeTreeStates(EnterStates: TChangeStates);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Class

TWorkerThread Class

Links

TWorkerThread Class

What do you think about this topic? Send feedback!
TWorkerThread.Create Constructor

TWorkerThread Class

Not documented.

Pascal

```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

Links

TWorkerThread Class

What do you think about this topic? Send feedback!
TWorkerThread.Destroy Destructor

TWorkerThread Class

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

Links
TWorkerThread Class

What do you think about this topic? Send feedback!
TWorkerThread.Execute Method

TWorkerThread Class

Not documented.

Pascal

```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

Links

TWorkerThread Class

What do you think about this topic? Send feedback!
**TWorkerThread.RemoveTree Method**

**TWorkerThread Class**

Not documented.

**Pascal**

```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

**Links**

TWorkerThread Class

*What do you think about this topic? Send feedback!*
TWriterHack Class

Classes

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

Class Hierarchy

File

VirtualTrees

Links

Classes

What do you think about this topic? Send feedback!
Functions

Functions | Virtual Treeview | Legend

These are all functions that are contained in this documentation.

Functions

AlphaBlend
  General purpose procedure to blend one bitmap to another.

DrawTextW
  Paint support procedure.

EnumerateVTClipboardFormats
  Not documented.

EnumerateVTClipboardFormats
  Not documented.

GetVTClipboardFormatDescription
  Not documented.

PrtStretchDrawDIB
  Not documented.

RegisterVTClipboardFormat
  Methods to register a certain clipboard format for a given tree class.

RegisterVTClipboardFormat
  Methods to register a certain clipboard format for a given tree class.

ShortenString
  General purpose routine to shorten a Unicode string to a given maximum size.

TreeFromNode
  General purpose routine to get the tree to which a node belongs.

Group
Virtual Treeview

Legend

Function

Links
Functions, Virtual Treeview, Legend

What do you think about this topic? Send feedback!
AlphaBlend Function

General purpose procedure to blend one bitmap to another.

Pascal

```pascal
procedure AlphaBlend(Source: HDC; Destination: HDC;
```

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

File
VirtualTrees

Links
Functions

What do you think about this topic? Send feedback!
**DrawTextW Function**

**Functions**

Paint support procedure.

**Pascal**

```
procedure DrawTextW(DC: HDC; lpString: PWideChar; nCount: Integer;
```

**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 accommodate 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

**File**

- VirtualTrees

**Links**

- Functions

---

*What do you think about this topic? Send feedback!*
EnumerateVTClipboardFormats Function

Not documented.

Pascal

```pascal
procedure EnumerateVTClipboardFormats(TreeClass: TVirtualTreeClass);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Functions

File

VirtualTrees

Links

Functions

What do you think about this topic? Send feedback!
EnumerateVTClipboardFormats Function

Not documented.

Pascal

```pascal
procedure EnumerateVTClipboardFormats(TreeClass: TVi
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Functions

File

VirtualTrees

Links

Functions

What do you think about this topic? Send feedback!
GetVTClipboardFormatDescription Function

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Pascal

```
function GetVTClipboardFormatDescription(AFormat: Word): Description;
```

Group
Functions

File
VirtualTrees

Links
Functions

What do you think about this topic? Send feedback!
PrtStretchDrawDIB Function

Functions

Not documented.

Pascal

```pascal
procedure PrtStretchDrawDIB(Canvas: TCanvas; DestRect: TRect; AB
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Functions

File

VirtualTrees

Links

Functions

What do you think about this topic? Send feedback!
RegisterVTClipboardFormat Function

Functions

Methods to register a certain clipboard format for a given tree class.

Pascal

```pascal
procedure RegisterVTClipboardFormat(AFormat: Word; T
function RegisterVTClipboardFormat(Description: string
```

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

File

VirtualTrees

Links

Functions

What do you think about this topic? Send feedback!
ShortenString Function

General purpose routine to shorten a Unicode string to a given maximum size.

Pascal

```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 course, 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 implicitly. 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 ellipsis on the correct side. The result is the left part of the string which fits into the given space plus the ellipsis.

Notes

It is assumed that the string really needs shortage. Check this in advance.

Group

Functions
TreeFromNode Function

General purpose routine to get the tree to which a node belongs.

Pascal

```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.
Functions

File
  VirtualTrees

Links
  Functions

What do you think about this topic? Send feedback!
Structs and Records

Enumerations | Virtual Treeview | Records | Legend

These are all structs and records that are contained in this documentation.

Enumerations

- TVTTooltipLineBreakStyle
  Not documented.

Group

Virtual Treeview

Records

- TBaseChunk
  Not documented.
- TBaseChunkBody
  Not documented.
- TCacheEntry
  Not documented.
- TChunkHeader
  Not documented.
- TClipboardFormatEntry
  Not documented.
- TClipboardFormatListEntry
  Not documented.
- THeaderPaintInfo
  Not documented.
- THitInfo
Not documented.

\textbf{TInternalStgMedium}
Not documented.

\textbf{TRealWMNCPaint}
Not documented.

\textbf{TSHDragImage}
Not documented.

\textbf{TToggleAnimationData}
Not documented.

\textbf{TVirtualNode}
Not documented.

\textbf{TVTHintData}
Not documented.

\textbf{TVTImageInfo}
Not documented.

\textbf{TVTPaintInfo}
Not documented.

\textbf{TVTReference}
Not documented.

\textbf{TWMPrint}
Not documented.

**Legend**

- \textbf{Struct}

**Links**

Enumerations, Virtual Treeview, Records, Legend

\textit{What do you think about this topic? Send feedback!}
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

File

VirtualTrees

Links

Structs and Records

What do you think about this topic? Send feedback!
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

File
VirtualTrees

Links
Structs and Records

What do you think about this topic? Send feedback!
TCacheEntry Record

Structs and Records

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

File

VirtualTrees

Links

Structs and Records

What do you think about this topic? Send feedback!
**TChunkHeader Record**

*Structs and Records*

Not documented.

**Pascal**

```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*

**File**

VirtualTrees

**Links**

*Structs and Records*

---

*What do you think about this topic? Send feedback!*
TClipboardFormatEntry Record

Not documented.

Pascal

```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

File

VirtualTrees

Links

Structs and Records

What do you think about this topic? Send feedback!
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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description: string;</td>
<td>The string used to register the format with Windows.</td>
</tr>
<tr>
<td>TreeClass: TVirtualTreeClass;</td>
<td>The tree class which supports rendering this format.</td>
</tr>
<tr>
<td>Priority: Cardinal;</td>
<td>Number which determines the order of formats used in IDataObject.</td>
</tr>
<tr>
<td>FormatEtc: TFormatEtc;</td>
<td>The definition of the format in the IDataObject.</td>
</tr>
</tbody>
</table>
Group
   Structs and Records

File
   VirtualTrees

Links
   Structs and Records

What do you think about this topic? Send feedback!
THeaderPaintInfo Record

Structs and Records

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

File

VirtualTrees
Links

Structs and Records

What do you think about this topic? Send feedback!
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

File
VirtualTrees

Links
Structs and Records

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Structs and Records

What do you think about this topic? Send feedback!
TRealWMNCPaint Record

Structs and Records

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

File

VirtualTrees

Links

Structs and Records

What do you think about this topic? Send feedback!
**TSHDragImage Record**

**Structs and Records**

Not documented.

**Pascal**

```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

**File**

VirtualTrees

**Links**

Structs and Records

What do you think about this topic? Send feedback!
TToggleAnimationData Record

Structs and Records

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expand: Boolean;</td>
<td>if true then expanding is in progress</td>
</tr>
<tr>
<td>Window: HWND;</td>
<td>copy of the tree's window handle</td>
</tr>
<tr>
<td>DC: HDC;</td>
<td>the DC of the window to erase unconvered parts</td>
</tr>
<tr>
<td>Brush: HBRUSH;</td>
<td>the brush to be used to erase uncovered parts</td>
</tr>
<tr>
<td>R: TRect;</td>
<td>the scroll rectangle</td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NodeHeight: Word;</td>
<td>height in pixels</td>
</tr>
<tr>
<td>States: TVirtualNodeStates;</td>
<td>states describing various properties of the node (expanded, initialized etc.)</td>
</tr>
<tr>
<td>Align: Byte;</td>
<td>line/button alignment</td>
</tr>
<tr>
<td>CheckState: TCheckState;</td>
<td>checked, pressed etc.</td>
</tr>
<tr>
<td>CheckType: TCheckType;</td>
<td>indicates which check type shall be used for this node</td>
</tr>
<tr>
<td>Dummy: Byte;</td>
<td>dummy value to fill DWORD boundary</td>
</tr>
<tr>
<td>Data: record;</td>
<td>this is a placeholder, each node gets extra data determined by NodeDataSize</td>
</tr>
</tbody>
</table>

**Group**
- Structs and Records

**File**
- VirtualTrees

**Links**
- Structs and Records

*What do you think about this topic? Send feedback!*
TVTHintData Record

Structs and Records

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HintRect: TRect;</td>
<td>used for draw trees only, string trees get the size from the hint string</td>
</tr>
<tr>
<td>DefaultHint: WideString;</td>
<td>used only if there is no node specific hint string available or a header hint is about to appear</td>
</tr>
</tbody>
</table>
HintText: WideString; set when size of the hint window is calculated

Group
   Structs and Records

File
   VirtualTrees

Links
   Structs and Records

What do you think about this topic? Send feedback!
**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**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index: Integer;</td>
<td>index in the associated image list</td>
</tr>
<tr>
<td>Ghosted: Boolean;</td>
<td>flag to indicate that the image must be drawn slightly lighter</td>
</tr>
</tbody>
</table>

**Group**

**Structs and Records**

**File**

**VirtualTrees**
What do you think about this topic? Send feedback!
TVTPaintInfo Record

Structs and Records

Not documented.

Pascal

```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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canvas: TCanvas;</td>
<td>the canvas to paint on</td>
</tr>
<tr>
<td>PaintOptions:</td>
<td>a copy of the paint options passed to PaintTree</td>
</tr>
<tr>
<td>TVTInternalPaintOptions;</td>
<td></td>
</tr>
</tbody>
</table>
### 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

**File**

- VirtualTrees

**Links**

- Structs and Records

---

*What do you think about this topic? Send feedback!*
TVTReference Record

Structs and Records

Not documented.

Pascal

```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

File

VirtualTrees

Links

Structs and Records

What do you think about this topic? Send feedback!
**TVTTTooltipLineBreakStyle Enumeration**

**Structs and Records**

Not documented.

**Pascal**

```pascal
TVTTTooltipLineBreakStyle = (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**

**File**

VirtualTrees

**Links**

**Structs and Records**

---

*What do you think about this topic? Send feedback!*
**TWMPrint Record**

**Structs and Records**

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

**File**

VirtualTrees

**Links**

Structs and Records

*What do you think about this topic? Send feedback!*
These are all types that are contained in this documentation.

Enumerations

- **TAddPopupItemType**
  
  Not documented.

- **TBlendMode**
  
  Not documented.

- **TChangeReason**
  
  Not documented.

- **TCheckImageKind**
  
  Determines which images should be used for checkboxes and radio buttons.

- **TCheckState**
  
  Returns the current state of a node's check box, radio button or node button.

- **TCheckType**
  
  Not documented.

- **TDragOperation**
  
  Not documented.

- **TDropMode**
  
  Not documented.

- **THeaderState**
  
  Not documented.

- **THintAnimationType**
  
  Not documented.

- **THitPosition**
  
  Not documented.
- **TItemEraseAction**
  Not documented.
- **TScrollBarStyle**
  Not documented.
- **TSortDirection**
  Not documented.
- **TVirtualNodeInitState**
  Not documented.
- **TVirtualNodeState**
  Not documented.
- **TVirtualTreeColumnStyle**
  Not documented.
- **TVSTTextSourceType**
  Not documented.
- **TVSTTextType**
  Not documented.
- **TVTAnimationOption**
  Not documented.
- **TVTAutoOption**
  Not documented.
- **TVTButtonFillMode**
  Determines how the interior of nodes buttons should be drawn.
- **TVTButtonStyle**
  Not documented.
- **TVTColumnOption**
  Not documented.
- **TVTDragImageKind**
  Not documented.
- **TVTDragMoveRestriction**
  Not documented.
- **TVTDragType**
  Not documented.
- **TVTDropSelectionMode**
  Not documented.
Not documented.

TVTHeaderColumnLayout
Not documented.

TVTHeaderOption
Not documented.

TVTHeaderPopupOption
Not documented.

TVTHeaderStyle
Not documented.

TVTHintMode
Not documented.

TVTImageInfoIndex
Not documented.

TVTImageKind
Not documented.

TVTIncrementalSearch
Not documented.

TVTInternalPaintOption
Not documented.

TVTLineMode
Not documented.

TVTLineStyle
Not documented.

TVTLineType
Not documented.

TVTMiscOption
Not documented.

TVTNodeAlignment
Not documented.

TVTNodeAttachMode
Not documented.

TVTPaintOption
Not documented.

TVTSearchDirection
Not documented.
TVTSearchStart
Not documented.

TVTSelectionOption
Not documented.

TVTStringOption
Not documented.

TVTUpdateState
Not documented.

Group
Virtual Treeview

Types

PCardinal
Not documented.

PClipboardFormatListEntry
Not documented.

PSHDragImage
Not documented.

PVirtualNode
Not documented.

PVTHintData
Not documented.

PVTReference
Not documented.

TAddHeaderPopupItemEvent
Not documented.

TAutoScrollInterval
Not documented.

TCache
Not documented.

TCardinalArray
Not documented.

TChangeStates
Not documented.

TColumnChangeEvent
Not documented.

TColumnIndex
Not documented.

TColumnPosition
Not documented.

TColumnsArray
Not documented.

TDragOperations
Not documented.

TFormatArray
Not documented.

TFormatEtcArray
Not documented.

TGetFirstNodeProc
Not documented.

TGetNextNodeProc
Not documented.

THeaderPaintElements
Not documented.

THeaderStates
Not documented.

THitPositions
Not documented.

TImageIndex
Not documented.

TIndexArray
Not documented.

TInternalStgMediumArray
Not documented.

TLineImage
Not documented.

TMagicID
Not documented.
- TMouseButtons
  Not documented.
- TNodeArray
  Not documented.
- TScrollDirections
  Not documented.
- TScrollUpdateOptions
  Not documented.
- TTreeOptionsClass
  Not documented.
- TVirtualNodeInitStates
  Not documented.
- TVirtualNodeStates
  Not documented.
- TVirtualTreeClass
  Not documented.
- TVirtualTreeColumnClass
  Not documented.
- TVirtualTreeColumnsClass
  Not documented.
- TVirtualTreeStates
  Not documented.
- TVSTGetTextEvent
  Not documented.
- TVSTNewTextEvent
  Not documented.
- TVSTShortenStringEvent
  Not documented.
- TVTAdvancedHeaderPaintEvent
  Not documented.
- TVTAfterCellPaintEvent
  Not documented.
- TVTAfterItemEraseEvent
  Not documented.
- TVTAfterItemPaintEvent
Not documented.

TVTAnimationCallback
Not documented.

TVTAnimationOptions
Not documented.

TVTAutoOptions
Not documented.

TVTBackgroundPaintEvent
Not documented.

TVTBeforeCellPaintEvent
Not documented.

TVTBeforeItemEraseEvent
Not documented.

TVTBeforeItemPaintEvent
Not documented.

TVTBias
Not documented.

TVTChangeEvent
Not documented.

TVTChangingEvent
Not documented.

TVTCheckChangingEvent
Not documented.

TVTColumnClickEvent
Not documented.

TVTColumnDblClickEvent
Not documented.

TVTColumnOptions
Not documented.

TVTCompareEvent
Not documented.

TVTCreateDataObjectEvent
Not documented.

TVTCreateDragManagerEvent
Not documented.
TVTCreateEditorEvent
Not documented.

TVTDragAllowedEvent
Not documented.

TVTDragDropEvent
Not documented.

TVTDragImageStates
Not documented.

TVTDragOverEvent
Not documented.

TVTDragHintEvent
Not documented.

TVTDrawNodeEvent
Not documented.

TVTEditCancelEvent
Not documented.

TVTEditChangeEvent
Not documented.

TVTEditChangingEvent
Not documented.

TVTFocusChangeEvent
Not documented.

TVTFocusChangingEvent
Not documented.

TVTFreeNodeEvent
Not documented.

TVTGetCursorEvent
Not documented.

TVTGetHeaderCursorEvent
Not documented.

TVTGetHintSizeEvent
Not documented.

TVTGetImageEvent
Not documented.

TVTGetLineStyleEvent
Not documented.

**TVTGetNodeDataSizeEvent**
Not documented.

**TVTGetNodeProc**
Not documented.

**TVTGetNodeWidthEvent**
Not documented.

**TVTGetUserClipboardFormatsEvent**
Not documented.

**TVTHeaderClass**
Not documented.

**TVTHeaderClickEvent**
Not documented.

**TVTHeaderDraggedEvent**
Not documented.

**TVTHeaderDraggedOutEvent**
Not documented.

**TVSTGetHintEvent**
Not documented.

**TVTHeaderDraggingEvent**
Not documented.

**TVTHeaderMouseEvent**
Not documented.

**TVTHeaderMouseMoveEvent**
Not documented.

**TVTHeaderNotifyEvent**
Not documented.

**TVTHeaderOptions**
Not documented.

**TVTHeaderPaintEvent**
Not documented.

**TVTHeaderPaintQueryElementsEvent**
Not documented.

**TVTHeaderPopupOptions**
Not documented.
TVTHelpContextEvent
Not documented.
TVTHotNodeChangeEvent
Not documented.
TVTIncrementalSearchEvent
Not documented.
TVTInitChildrenEvent
Not documented.
TVTInitNodeEvent
Not documented.
TVTInternalPaintOptions
Not documented.
TVTKeyActionEvent
Not documented.
TVTMeasureItemEvent
Not documented.
TVTMiscOptions
Not documented.
TVTNodeCopiedEvent
Not documented.
TVTNodeCopyingEvent
Not documented.
TVTNodeMovedEvent
Not documented.
TVTNodeMovingEvent
Not documented.
TVTPaintEvent
Not documented.
TVTPaintOptions
Not documented.
TVTPaintText
Not documented.
TVTPopupEvent
Not documented.
TVTRenderOLEDataEvent
Not documented.

TVTSaveNodeEvent
Not documented.

TVTScrollEvent
Not documented.

TVTScrollIncrement
Not documented.

TVTSelectionOptions
Not documented.

TVTStateChangeEvent
Not documented.

TVTStringOptions
Not documented.

TVTStructureChangeEvent
Not documented.

TVTTTransparency
Not documented.

TVTUpdatingEvent
Not documented.

TWMContextMenu
Not documented.

TWMPrintClient
Not documented.

TVTGetCellIsEmptyEvent
Not documented.

TVTGetImageExEvent
Not documented.

TVTMenuItem
Not documented.

TVTScrollbarShowEvent
Not documented.

Legend
What do you think about this topic? Send feedback!
**PCardinal Type**

**Types**

Not documented.

**Pascal**

```pascal
PCardinal = ^Cardinal;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
PClipboardFormatListEntry Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
PSHDragImage Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**PVirtualNode Type**

**Types**

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

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
PVTHintData Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**PVTReference Type**

**Types**

Not documented.

**Pascal**

```pascal
PVTReference = ^TVTReference;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- Types

**File**

- VirtualTrees

**Links**

- Types

---

*What do you think about this topic? Send feedback!*
# TAddHeaderPopupItemEvent Type

## Types

Not documented.

## Pascal

```pascal
tAddHeaderPopupItemEvent = procedure (const Sender: TBaseVirtualTree);
```

## Description

Use other resources like the news group or the Delphi Gems message board to find a description.

## Group

Types

## File

VTHeaderPopup

## Links

Types

*What do you think about this topic? [Send feedback]*
**TAutoScrollInterval Type**

**Types**

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

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TCache Type**

**Types**

Not documented.

**Pascal**

```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

**File**

- VirtualTrees

**Links**

- Types

---

*What do you think about this topic? Send feedback!*
TCardinalArray Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TChangeStates Type

Types

Not documented.

Pascal

TChangeStates = set of (csStopValidation, csUseCache)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TColumnChangeEvent Type

Types

Not documented.

Pascal

```pascal
TColumnChangeEvent = procedure (const Sender: TBaseV
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VTHHeaderPopup

Links

Types

What do you think about this topic? Send feedback!
TColumnIndex Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TColumnPosition Type**

*Types*

Not documented.

*Pascal*

```pascal
TColumnPosition = type Cardinal;
```

*Description*

Use other resources like the news group or the Delphi Gems message board to find a description.

*Group*

*Types*

*File*

VirtualTrees

*Links*

*Types*

What do you think about this topic? Send feedback!
TColumnsArray Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TDragOperations Type**

**Types**

Not documented.

**Pascal**

```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**

**File**

VirtualTrees

**Links**

**Types**

*What do you think about this topic? Send feedback!*
**TFormatArray Type**

*Types*

Not documented.

*Pascal*

```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

*File*

VirtualTrees

*Links*

Types

*What do you think about this topic?* Send feedback!
TFormatEtcArray Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TGetFirstNodeProc Type**

**Types**

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

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
TGetNextNodeProc Type

Types

Not documented.

Pascal

TGetNextNodeProc = function (Node: PVirtualNode): PVirtualNode;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
THeaderPaintElements Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**THeaderStates Type**

**Types**

Not documented.

**Pascal**

```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

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**THitPositions Type**

**Types**

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

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic?* Send feedback!
TImageIndex Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TIndexArray Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TInternalStgMediumArray Type**

**Types**

Not documented.

**Pascal**

```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

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? Send feedback!*
**TLineImage Type**

Types

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

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
TMagicID Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TMouseButtons Type**

**Types**

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

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
TNodeArray Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TScrollDirections Type

**Types**

Not documented.

**Pascal**

```pascal
TScrollDirections = set of (sdLeft, sdUp, sdRight, )
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TScrollUpdateOptions Type**

Types

Not documented.

Pascal

```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

File

VirtualTrees

Links

Types

*What do you think about this topic? Send feedback!*
**TTreeOptionsClass Type**

**Types**

Not documented.

**Pascal**

```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

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
### TVirtualNodeInitStates Type

**Types**

Not documented.

**Pascal**

```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

**File**

- VirtualTrees

**Links**

- Types

*What do you think about this topic? Send feedback!*
TVirtualNodeStates Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVirtualTreeClass Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVirtualTreeColumnClass Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
# TVirtualTreeColumnsClass Type

**Types**

Not documented.

**Pascal**

```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

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? [Send feedback]*
TVirtualTreeStates Type

Types

Not documented.

Pascal

```
TVirtualTreeStates = set of ( tsCancelHintAnimation,
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVSTGetTextEvent Type

Types

Not documented.

Pascal

```
TVSTGetTextEvent = procedure (Sender: TBaseVirtualTree);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVSTNewTextEvent Type

Types

Not documented.

Pascal

TVSTNewTextEvent = procedure (Sender: TBaseVirtualTree);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVSTShortenStringEvent Type**

*Types*

Not documented.

*Pascal*

```pascal
TVSTShortenStringEvent = procedure (Sender: TBaseVir
```

*Description*

Use other resources like the news group or the Delphi Gems message board to find a description.

*Group*

Types

*File*

VirtualTrees

*Links*

Types

*What do you think about this topic? Send feedback!*
**TVTAdvancedHeaderPaintEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTAdvancedHeaderPaintEvent = procedure (Sender: TVT)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? Send feedback!*
**TVTAfterCellPaintEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTAfterCellPaintEvent = procedure (Sender: TBaseVir
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? Send feedback!*
**TVTAfterItemEraseEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTAfterItemEraseEvent = procedure (Sender: TBaseVir
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVTAfterItemPaintEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTAfterItemPaintEvent = procedure (Sender: TBaseVir
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
TVTAnimationCallback Type

Types

Not documented.

Pascal

TVTAnimationCallback = function (Step, StepSize: Integer;
Data: Pointer): Boolean

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTAnimationOptions Type**

**Types**

Not documented.

**Pascal**

```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

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
TVTAutoOptions Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTBackgroundPaintEvent Type

Types
Not documented.

Pascal

```
TVTBackgroundPaintEvent = procedure (Sender: TBaseVi
```

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Group
Types

File
VirtualTrees

Links
Types

What do you think about this topic? Send feedback!
TVTBeforeCellPaintEvent Type

Types

Not documented.

Pascal

```pascal
TVTBeforeCellPaintEvent = procedure (Sender: TBaseVi
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTBeforeItemEraseEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTBeforeItemEraseEvent = procedure (Sender: TBaseVi
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
TVTBeforeItemPaintEvent Type

Types

Not documented.

Pascal

| TVTBeforeItemPaintEvent = procedure (Sender: TBaseVi |

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTBias Type

Types

Not documented.

Pascal

```pascal
TVTBias = -128..127;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTChangeEvent Type

Types

Not documented.

Pascal

```pascal
TVTChangeEvent = procedure (Sender: TBaseVirtualTree)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTChangingEvent Type**

**Types**

Not documented.

**Pascal**

```
TVTChangingEvent = procedure (Sender: TBaseVirtualTree)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- Types

**File**

- VirtualTrees

**Links**

- Types

---

What do you think about this topic? Send feedback!
**TVTCheckChangingEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTCheckChangingEvent = procedure (Sender: TBaseVirtualTree);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? Send feedback!*
TVTColumnClickEvent Type

Types

Not documented.

Pascal

```
TVTColumnClickEvent = procedure(Sender: TBaseVirtualTree)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTColumnDblClickEvent Type

Types

Not documented.

Pascal

TVTColumnDblClickEvent = procedure (Sender: TBaseVir

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTColumnOptions Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTCompareEvent Type

Types

Not documented.

Pascal

```
TVTCompareEvent = procedure (Sender: TBaseVirtualTree)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTCreateDataObjectEvent Type

Types

Not documented.

Pascal

```
TVTCreateDataObjectEvent = procedure (Sender: TBaseV
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTCreatedDragManagerEvent Type

Types

Not documented.

Pascal

TVTCreatedDragManagerEvent = procedure (Sender: TBase)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTCreateEditorEvent Type

Types
Not documented.

Pascal

TVTCreateEditorEvent = procedure (Sender: TBaseVirtualTree)

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Group
Types

File
VirtualTrees

Links
Types

What do you think about this topic? Send feedback!
TVTDragAllowedEvent Type

Types

Not documented.

Pascal

```
TVTDragAllowedEvent = procedure (Sender: TBaseVirtualTree) ;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTDragDropEvent Type**

**Types**

Not documented.

**Pascal**

```
TVTDragDropEvent = procedure (Sender: TBaseVirtualTree)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVTDragImageStates Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTDragImageStates = set of ( disHidden, disInDrag, ...
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVTDragOverEvent Type**

*Types*

Not documented.

*Pascal*

```pascal
TVTDragOverEvent = procedure (Sender: TBaseVirtualTree)
```

*Description*

Use other resources like the news group or the Delphi Gems message board to find a description.

*Group*

Types

*File*

VirtualTrees

*Links*

Types

*What do you think about this topic? Send feedback!*
**TVTDrawHintEvent Type**

**Types**

Not documented.

**Pascal**

```
TVTDrawHintEvent = procedure (Sender: TBaseVirtualTree)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVTDrawNodeEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTDrawNodeEvent = procedure (Sender: TBaseVirtualTree)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVTEditCancelEvent Type**

*Types*

Not documented.

*Pascal*

```
TVTEditCancelEvent = procedure (Sender: TBaseVirtual
```

*Description*

Use other resources like the news group or the Delphi Gems message board to find a description.

*Group*

Types

*File*

VirtualTrees

*Links*

Types

What do you think about this topic? Send feedback!
TVTEditChangeEvent Type

Types

Not documented.

Pascal

```
TVTEditChangeEvent = procedure (Sender: TBaseVirtual
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTEditChangingEvent Type

Types

Not documented.

Pascal

```pascal
TVTEditChangingEvent = procedure (Sender: TBaseVirtualTree)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTFocusChangeEvent Type

Types

Not documented.

Pascal

TVTFocusChangeEvent = procedure (Sender: TBaseVirtualTree)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTFocusChangingEvent Type

Types

Not documented.

Pascal

\texttt{TVTFocusChangingEvent = procedure (Sender: TBaseVirt)}

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

*What do you think about this topic? Send feedback!*
**TVTFreeNodeEvent Type**

Types

Not documented.

Pascal

```
TVTFreeNodeEvent = procedure (Sender: TBaseVirtualTree)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

*What do you think about this topic? Send feedback!*
TVTGetCursorEvent Type

Types

Not documented.

Pascal

```pascal
TVTGetCursorEvent = procedure (Sender: TBaseVirtualT
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTGetHeaderCursorEvent Type**

<table>
<thead>
<tr>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not documented.</td>
</tr>
</tbody>
</table>

**Pascal**

```pascal
TVTGetHeaderCursorEvent = procedure (Sender: TVTHeader) |
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- Types

**File**

- VirtualTrees

**Links**

- Types

---

*What do you think about this topic? Send feedback!*
**TVTGetHintSizeEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTGetHintSizeEvent = procedure (Sender: TBaseVirtual...
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? Send feedback!*
**TVTGetImageEvent Type**

*Types*

Not documented.

*Pascal*

```pascal
TVTGetImageEvent = procedure (Sender: TBaseVirtualTree)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

*Types*

**File**

VirtualTrees

**Links**

*Types*

---

*What do you think about this topic? Send feedback!*
**TVTGetLineStyleEvent Type**

**Types**

Not documented.

**Pascal**

```
TVTGetLineStyleEvent = procedure (Sender: TBaseVirtualTree)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? Send feedback!*
**TVTGetNodeDataSizeEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTGetNodeDataSizeEvent = procedure (Sender: TBaseVi
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
TVTGetNodeProc Type

Types

Not documented.

Pascal

```
TVTGetNodeProc = procedure (Sender: TBaseVirtualTree)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTGetNodeWidthEvent Type

Types

Not documented.

Pascal

TVTGetNodeWidthEvent = procedure (Sender: TBaseVirtualTree)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTGetUserClipboardFormatsEvent**

**Types**

<table>
<thead>
<tr>
<th>Types</th>
<th>TVTGetUserClipboardFormatsEvent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not documented.</td>
<td></td>
</tr>
</tbody>
</table>

**Pascal**

```pascal
TVTGetUserClipboardFormatsEvent = procedure (Sender: |
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

*Types*

**File**

VirtualTrees

**Links**

*Types*

*What do you think about this topic? Send feedback!*
TVTHHeaderClass Type

Types

Not documented.

Pascal

TVTHHeaderClass = class of TVTHHeader;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTHeaderClickEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTHeaderClickEvent = procedure (Sender: TVTHeader;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
TVTHeaderDraggedEvent Type

Types

Not documented.

Pascal

\[
\text{TVTHeaderDraggedEvent} = \text{procedure} (\text{Sender: TVTHeader})
\]

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTHeaderDraggedOutEvent Type

Types

Not documented.

Pascal

TVTHeaderDraggedOutEvent = procedure (Sender: TVThea) ...

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVSTGetHintEvent Type

Types

Not documented.

Pascal

TVSTGetHintEvent = procedure (Sender: TBaseVirtualTree)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTHeaderDraggingEvent Type

Types

Not documented.

Pascal

```pascal
TVTHeaderDraggingEvent = procedure (Sender: TVTHeader
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTHeaderMouseEvent Type

Types

Not documented.

Pascal

TVTHeaderMouseEvent = procedure (Sender: TVTHeader;)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTHeaderMouseMoveEvent Type

Not documented.

Pascal

```pascal
TVTHeaderMouseMoveEvent = procedure (Sender: TVTHeader)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTHeaderNotifyEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTHeaderNotifyEvent = procedure (Sender: TVTHeader;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVTHeaderOptions Type**

**Types**

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

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? Send feedback!*
**TVTHeaderPaintEvent** Type

**Types**

Not documented.

**Pascal**

```pascal
TVTHeaderPaintEvent = procedure (Sender: TVTHeader);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVTHeaderPaintQueryElementsEvent**

**Type**

- **Types**
  - Not documented.

**Pascal**

```pascal
TVTHeaderPaintQueryElementsEvent = procedure (Sender
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- **Types**

**File**

- VirtualTrees

**Links**

- **Types**

---

*What do you think about this topic? Send feedback!*
**TVTHHeaderPopupOptions Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTHHeaderPopupOptions = set of TVTHHeaderPopupOption;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

**Types**

**File**

**VTHHeaderPopup**

**Links**

**Types**

*What do you think about this topic? Send feedback!*
TVTHelpContextEvent Type

Types

Not documented.

Pascal

TVTHelpContextEvent = procedure (Sender: TBaseVirtualTree)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTHotNodeChangeEvent Type

Types

Not documented.

Pascal

TVTHotNodeChangeEvent = procedure (Sender: TBaseVirtualTree)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTIncrementalSearchEvent Type

**Types**

Not documented.

**Pascal**

```pascal
TVTIncrementalSearchEvent = procedure (Sender: TBase)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVTInitChildrenEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTInitChildrenEvent = procedure (Sender: TBaseVirtualTreeNode);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

What do you think about this topic? Send feedback!
TVTInitNodeEvent Type

Types

Not documented.

Pascal

\[
\text{TVTInitNodeEvent} = \text{procedure} (\text{Sender: TBaseVirtualTree})
\]

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTInternalPaintOptions Type

Types

Not documented.

Pascal

```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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTKeyActionEvent Type

Types

Not documented.

Pascal

TVTKeyActionEvent = procedure (Sender: TBaseVirtualT

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTMeasureItemEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTMeasureItemEvent = procedure (Sender: TBaseVirtualTree)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- Types

**File**

- VirtualTrees

**Links**

- Types

*What do you think about this topic? Send feedback!*
TVTMiscOptions Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTNodeCopiedEvent Type

Types

Not documented.

Pascal

\[
\text{TVTNodeCopiedEvent} = \text{procedure (Sender: TBaseVirtualTree)}
\]

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTNodeCopyingEvent Type

Types

Not documented.

Pascal

```
TVTNodeCopyingEvent = procedure (Sender: TBaseVirtualTree); 
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTNodeMovedEvent Type**

**Types**

Not documented.

**Pascal**

```pascal
TVTNodeMovedEvent = procedure (Sender: TBaseVirtualT
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

What do you think about this topic? Send feedback!
TVTNodeMovingEvent Type

Types

Not documented.

Pascal

TVTNodeMovingEvent = procedure (Sender: TBaseVirtual

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTPaintEvent Type

Types

Not documented.

Pascal

```
TVTPaintEvent = procedure (Sender: TBaseVirtualTree);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTPaintOptions** Type

**Types**

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

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic?*  [Send feedback!]*What do you think about this topic?*  [Send feedback!]
TVTPaintText Type

Types

Not documented.

Pascal

```
TVTPaintText = procedure (Sender: TBaseVirtualTree;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTPopupEvent Type

Types

Not documented.

Pascal

TVTPopupEvent = procedure (Sender: TBaseVirtualTree;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTRenderOLEDataEvent Type

Types

Not documented.

Pascal

TVTRenderOLEDataEvent = procedure (Sender: TBaseVirtualTree)

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTSaveNodeEvent** Type

**Types**

Not documented.

**Pascal**

```pascal
TVTSaveNodeEvent = procedure (Sender: TBaseVirtualTree)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? Send feedback!*
TVTScrollEvent Type

Not documented.

Pascal

```
TVTScrollEvent = procedure (Sender: TBaseVirtualTree)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTScrollIncrement Type

Types
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

File
VirtualTrees

Links
Types

What do you think about this topic? Send feedback!
TVTSelectionOptions Type

Types

Not documented.

Pascal

```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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
# TVTStateChangeEvent Type

**Types**

<table>
<thead>
<tr>
<th>Types</th>
<th>TVTSelectionOptions Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TVTStringOptions Type</td>
</tr>
</tbody>
</table>

**Not documented.**

**Pascal**

```
TVTStateChangeEvent = procedure (Sender: TBaseVirtualTree)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVTStringOptions Type**

**Types**

Not documented.

**Pascal**

```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**

**File**

VirtualTrees

**Links**

**Types**


*What do you think about this topic? Send feedback!*
**TVTStructureChangeEvent Type**

*Types*

Not documented.

*Pascal*

```pascal
TVTStructureChangeEvent = procedure (Sender: TBaseVi
```

*Description*

Use other resources like the news group or the Delphi Gems message board to find a description.

*Group*

*Types*

*File*

VirtualTrees

*Links*

*Types*

*What do you think about this topic? Send feedback!*
TVTTTransparency Type

Types

Not documented.

Pascal

```
TVTTTransparency = 0..255;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTUpdatingEvent Type**

Types

Not documented.

Pascal

```pascal
TVTUpdatingEvent = procedure (Sender: TBaseVirtualTree)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

*What do you think about this topic? Send feedback!*
**TWMContextMenu Type**

**Types**

Not documented.

**Pascal**

```pascal
TWMContextMenu = TWMMouse;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

---

*What do you think about this topic? Send feedback!*
TWMPrintClient Type

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TAddPopupItemType Enumeration**

**Types**

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

**File**

VTHHeaderPopup

**Links**

Types

*What do you think about this topic? Send feedback!*
TBlendMode Enumeration

Types

Not documented.

Pascal

TBlendMode = (bmConstantAlpha, bmPerPixelAlpha, bmMasterAlpha);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bmConstantAlpha</td>
<td>apply given constant alpha</td>
</tr>
<tr>
<td>bmPerPixelAlpha</td>
<td>use alpha value of the source pixel</td>
</tr>
<tr>
<td>bmMasterAlpha</td>
<td>use alpha value of source pixel and multiply it with the constant alpha value</td>
</tr>
<tr>
<td>bmConstantAlphaAndColor</td>
<td>blend the destination color with the given constant color and the constant alpha value</td>
</tr>
</tbody>
</table>

Group

Types

File

VirtualTrees
What do you think about this topic? Send feedback!
TVTGetCellIsEmptyEvent Type

Types
Not documented.

Pascal

TVTGetCellIsEmptyEvent = procedure (Sender: TBaseVir

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Group
Types

File
VirtualTrees

Links
Types

What do you think about this topic? Send feedback!
TChangeReason Enumeration

Types

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crIgnore</td>
<td>used as placeholder</td>
</tr>
<tr>
<td>crAccumulated</td>
<td>used for delayed changes</td>
</tr>
<tr>
<td>crChildAdded</td>
<td>one or more child nodes have been added</td>
</tr>
<tr>
<td>crChildDeleted</td>
<td>one or more child nodes have been deleted</td>
</tr>
<tr>
<td>crNodeAdded</td>
<td>a node has been added</td>
</tr>
<tr>
<td>crNodeCopied</td>
<td>a node has been duplicated</td>
</tr>
<tr>
<td>crNodeMoved</td>
<td>a node has been moved to a new place</td>
</tr>
</tbody>
</table>

Group

Types
File
VirtualTrees

Links
Types

What do you think about this topic? Send feedback!
**TCheckImageKind Enumeration**

**Types**

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.

- **Dark check images**
- **Light check images**
- **Dark tick images**
- **Light tick images**
- **Flat images**
- **Windows XP images**
- **System tick images**
- **Flat system check images**

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)

Radio buttons:

- uncheck normal (ckRadioUncheckedNormal)
- unchecked hot (ckRadioUncheckedHot)
- unchecked pressed (ckRadioUncheckedPressed)
- unchecked disabled (ckRadioUncheckedDisabled)
- checked normal (ckRadioCheckedNormal)
- checked hot (ckRadioCheckedHot)
- checked pressed (ckRadioCheckedPressed)
- checked disabled (ckRadioCheckedDisabled)

Check boxes:
- unchecked normal (ckCheckUncheckedNormal)
- unchecked hot (ckCheckUncheckedHot)
- unchecked pressed (ckCheckUncheckedPressed)
- unchecked disabled (ckCheckUncheckedDisabled)
- checked normal (ckCheckCheckedNormal)
- checked hot (ckCheckCheckedHot)
- checked pressed (ckCheckCheckedPressed)
- checked disabled (ckCheckCheckedDisabled)
- mixed normal (ckCheckMixedNormal)
- mixed hot (ckCheckMixedHot)
- mixed pressed (ckCheckMixedPressed)
- mixed disabled (ckCheckMixedDisabled)

Node buttons:

- button normal (ckButtonNormal)
- button hot (ckButtonHot)
- button pressed (ckButtonPressed)
- button disabled (ckButtonDisabled)

Members

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ckLightCheck</td>
<td>gray cross</td>
</tr>
<tr>
<td>ckDarkCheck</td>
<td>black cross</td>
</tr>
<tr>
<td>ckLightTick</td>
<td>gray tick mark</td>
</tr>
<tr>
<td>ckDarkTick</td>
<td>black tick mark</td>
</tr>
<tr>
<td>ckFlat</td>
<td>flat images (no 3D border)</td>
</tr>
<tr>
<td>ckXP</td>
<td>Windows XP style</td>
</tr>
<tr>
<td>ckCustom</td>
<td>application defined check images</td>
</tr>
<tr>
<td>ckSystem</td>
<td>System defined check images.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>ckSystemFlat</td>
<td>Flat system defined check images.</td>
</tr>
</tbody>
</table>

**Group**

**Types**

**File**

**VirtualTrees**

**Links**

**Types**

*What do you think about this topic? Send feedback!*
TCheckState Enumeration

Types

Returns the current state of a node's check box, radio button or node button.

Pascal

```pascal
TCheckState = (csUncheckedNormal, csUncheckedPressed)
```

Description

The check states include both, transient and fluent (temporary) states. The only temporary state defined so far is the pressed state.

Members

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>csUncheckedNormal</td>
<td>unchecked and not pressed</td>
</tr>
<tr>
<td>csUncheckedPressed</td>
<td>unchecked and pressed</td>
</tr>
<tr>
<td>csCheckedNormal</td>
<td>checked and not pressed</td>
</tr>
<tr>
<td>csCheckedPressed</td>
<td>checked and pressed</td>
</tr>
<tr>
<td>csMixedNormal</td>
<td>3-state check box and not pressed</td>
</tr>
<tr>
<td>csMixedPressed</td>
<td>3-state check box and pressed</td>
</tr>
</tbody>
</table>
What do you think about this topic? Send feedback!
TCheckType Enumeration

Types

Not documented.

Pascal

TCheckType = (ctNone, ctTriStateCheckBox, ctCheckBox);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TDragOperation Enumeration

Types

Not documented.

Pascal

```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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTGetImageExEvent Type

Types

Not documented.

Pascal

```pascal
TVTGetImageExEvent = procedure (Sender: TBaseVirtual
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TDropMode Enumeration**

*Types*

Not documented.

*Pascal*

```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

*File*

VirtualTrees

*Links*

Types

What do you think about this topic? Send feedback!
THHeaderState Enumeration

Types

Not documented.

Pascal

THHeaderState = (hsAutoSizing, hsDragging, hsDragPending);

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsAutoSizing</td>
<td>auto size chain is in progress, do not trigger again on WM_SIZE</td>
</tr>
<tr>
<td>hsDragging</td>
<td>header dragging is in progress (only if enabled)</td>
</tr>
<tr>
<td>hsDragPending</td>
<td>left button is down, user might want to start dragging a column</td>
</tr>
<tr>
<td>hsLoading</td>
<td>The header currently loads from stream, so updates are not necessary.</td>
</tr>
<tr>
<td>hsTracking</td>
<td>column resizing is in progress</td>
</tr>
<tr>
<td>hsTrackPending</td>
<td>left button is down, user might want to start resize a column</td>
</tr>
</tbody>
</table>
Types

File
   VirtualTrees

Links
   Types

What do you think about this topic? Send feedback!
**THintAnimationType Enumeration**

*Types*

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hatNone</td>
<td>no animation at all, just display hint/tooltip</td>
</tr>
<tr>
<td>hatFade</td>
<td>fade in the hint/tooltip, like in Windows 2000</td>
</tr>
<tr>
<td>hatSlide</td>
<td>slide in the hint/tooltip, like in Windows 98</td>
</tr>
<tr>
<td>hatSystemDefault</td>
<td>use what the system is using (slide for Win9x, slide/fade for Win2K+, depends on settings)</td>
</tr>
</tbody>
</table>

*Pascal*

```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*

File

VirtualTrees
What do you think about this topic? Send feedback!
THitPosition Enumeration

Types

Not documented.

Pascal

```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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hiAbove</td>
<td>above the client area (if relative) or the absolute tree area</td>
</tr>
<tr>
<td>hiBelow</td>
<td>below the client area (if relative) or the absolute tree area</td>
</tr>
<tr>
<td>hiNowhere</td>
<td>no node is involved (possible only if the tree is not as tall as the client area)</td>
</tr>
<tr>
<td>hiOnItem</td>
<td>on the bitmaps/buttons or label associated with an item</td>
</tr>
<tr>
<td>hiOnItemButton</td>
<td>on the button associated with an item</td>
</tr>
<tr>
<td>hiOnItemCheckbox</td>
<td>on the checkbox if enabled</td>
</tr>
<tr>
<td>hiOnItemIndent</td>
<td>in the indentation area in front of a node</td>
</tr>
<tr>
<td>hiOnItemLabel</td>
<td>on the normal text area associated with an item</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>hiOnItemLeft</td>
<td>when right aligned or centered)</td>
</tr>
<tr>
<td>hiOnItemRight</td>
<td>if left aligned or centered)</td>
</tr>
<tr>
<td>hiOnNormalIcon</td>
<td>on the &quot;normal&quot; image</td>
</tr>
<tr>
<td>hiOnStateIcon</td>
<td>on the state image</td>
</tr>
<tr>
<td>hiToLeft</td>
<td>to the left of the client area (if relative) or the absolute tree area</td>
</tr>
<tr>
<td>hiToRight</td>
<td>to the right of the client area (if relative) or the absolute tree area</td>
</tr>
</tbody>
</table>

**Group**

**Types**

**File**

**VirtualTrees**

**Links**

**Types**

---

*What do you think about this topic? Send feedback!*
TItemEraseAction Enumeration

Types

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eaColor</td>
<td>Use the provided color to erase the background instead the one of the tree.</td>
</tr>
<tr>
<td>eaDefault</td>
<td>The tree should erase the item's background (bitmap or solid).</td>
</tr>
<tr>
<td>eaNone</td>
<td>Let the application paint the background.</td>
</tr>
</tbody>
</table>

Group

Types

File

VirtualTrees

Links
Types

What do you think about this topic? Send feedback!
TScrollBarStyle Enumeration

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TSortDirection Enumeration**

**Types**

Not documented.

**Pascal**

```pascal
TSortDirection = (sdAscending, sdDescending);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

What do you think about this topic? Send feedback!
**TVirtualNodeInitState** Enumeration

**Types**

Not documented.

**Pascal**


**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

What do you think about this topic? Send feedback!
TVirtualNodeState Enumeration

Types

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vsInitialized</td>
<td>Set after the node has been initialized.</td>
</tr>
<tr>
<td>vsChecking</td>
<td>Node's check state is changing, avoid propagation.</td>
</tr>
<tr>
<td>vsCutOrCopy</td>
<td>Node is selected as cut or copy and paste source.</td>
</tr>
<tr>
<td>vsDisabled</td>
<td>Set if node is disabled.</td>
</tr>
<tr>
<td>vsDeleting</td>
<td>Set when the node is about to be freed.</td>
</tr>
<tr>
<td>vsExpanded</td>
<td>Set if the node is expanded.</td>
</tr>
<tr>
<td>vsHasChildren</td>
<td>Indicates the presence of child nodes without actually setting them.</td>
</tr>
<tr>
<td><strong>vsVisible</strong></td>
<td>Indicate whether the node is visible or not (independant of the expand states of its parents).</td>
</tr>
<tr>
<td><strong>vsSelected</strong></td>
<td>Set if the node is in the current selection.</td>
</tr>
<tr>
<td><strong>vsInitialUserData</strong></td>
<td>Set if (via AddChild or InsertNode) initial user data has been set which requires OnFreeNode.</td>
</tr>
<tr>
<td><strong>vsAllChildrenHidden</strong></td>
<td>Set if vsHasChildren is set and no child node has the vsVisible flag set.</td>
</tr>
<tr>
<td><strong>vsClearing</strong></td>
<td>Don't register structure change event.</td>
</tr>
<tr>
<td><strong>vsMultiline</strong></td>
<td>Node text is wrapped at the cell boundaries instead of being shorted.</td>
</tr>
<tr>
<td><strong>vsHeightMeasured</strong></td>
<td>Node height has been determined and does not need a recalculation.</td>
</tr>
</tbody>
</table>

**Group**

**Types**

**File**

VirtualTrees

**Links**

**Types**

*What do you think about this topic? Send feedback!*
**TVirtualTreeColumnStyle Enumeration**

**Types**

Not documented.

**Pascal**

```pascal
TVirtualTreeColumnStyle = (vsText, vsOwnerDraw);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVSTTextSourceType** Enumeration

**Types**

Not documented.

**Pascal**

```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**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tstAll</td>
<td>Initialization is done on the fly.</td>
</tr>
<tr>
<td>tstInitialized</td>
<td>Only initialized nodes are rendered.</td>
</tr>
<tr>
<td>tstSelected</td>
<td>Only selected nodes are rendered.</td>
</tr>
<tr>
<td>tstCutCopySet</td>
<td>Only nodes currently marked as being in the cut/copy clipboard set are rendered.</td>
</tr>
<tr>
<td>tstVisible</td>
<td>Only visible nodes are rendered.</td>
</tr>
</tbody>
</table>

**Group**

Types

**File**
### TVSTTextType Enumeration

**Types**

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**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttNormal</td>
<td>normal label of the node, this is also the text which can be edited</td>
</tr>
<tr>
<td>ttStatic</td>
<td>static (non-editable) text after the normal text</td>
</tr>
</tbody>
</table>

**Group**

Types

**File**

VirtualTrees

**Links**

Types

---

What do you think about this topic? Send feedback!
TVTAnimationOption Enumeration

Types

Not documented.

Pascal

```pascal
TVTAnimationOption = (toAnimatedToggle);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>toAnimatedToggle</td>
<td>Expanding and collapsing a node is animated (quick window scroll).</td>
</tr>
</tbody>
</table>

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTAutoOption Enumeration

Types

Not documented.

Pascal

```
TVTAutoOption = (toAutoDropExpand, toAutoExpand, ...
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>toAutoDropExpand</td>
<td>Expand node if it is the drop target for more than certain time.</td>
</tr>
<tr>
<td>toAutoExpand</td>
<td>Nodes are expanded (collapsed) when getting (losing) the focus.</td>
</tr>
<tr>
<td>toAutoScroll</td>
<td>Scroll if mouse is near the border while dragging or selecting.</td>
</tr>
<tr>
<td>toAutoScrollOnExpand</td>
<td>Scroll as many child nodes in view as possible after expanding a node.</td>
</tr>
<tr>
<td>toAutoSort</td>
<td>Sort tree when Header.SortColumn or Header.SortDirection change</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>toAutoSpanColumns</td>
<td>Large entries continue into next column(s) if there's no text in them (no clipping).</td>
</tr>
<tr>
<td>toAutoTristateTracking</td>
<td>Checkstates are automatically propagated for tri state check boxes.</td>
</tr>
<tr>
<td>toAutoHideButtons</td>
<td>Node buttons are hidden when there are child nodes, but all are invisible.</td>
</tr>
<tr>
<td>toAutoDeleteMovedNodes</td>
<td>Delete nodes which were moved in a drag operation (if not directed otherwise).</td>
</tr>
<tr>
<td>toDisableAutoscrollOnFocus</td>
<td>Disable scrolling a column entirely into view if it gets focused.</td>
</tr>
<tr>
<td>toAutoChangeScale</td>
<td>Change default node height automatically if the system's font scale is set to big fonts.</td>
</tr>
<tr>
<td>toAutoFreeOnCollapse</td>
<td>Frees any child node after a node has been collapsed (HasChildren flag stays there).</td>
</tr>
</tbody>
</table>

**Group**

- **Types**

**File**

- **VirtualTrees**

**Links**

- **Types**
What do you think about this topic? Send feedback!
**TVTButtonFillMode Enumeration**

**Types**

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fmTreeColor</td>
<td>solid color, uses the tree's background color</td>
</tr>
<tr>
<td>fmWindowColor</td>
<td>solid color, uses c1Window</td>
</tr>
<tr>
<td>fmShaded</td>
<td>color gradient, Windows XP style (legacy code, use toThemeAware on Windows XP instead)</td>
</tr>
<tr>
<td>fmTransparent</td>
<td>transparent color, use the item's background color</td>
</tr>
</tbody>
</table>

Group

Types

File

VirtualTrees

Links

Types
What do you think about this topic? Send feedback!
TVTButtonStyle Enumeration

Types

Not documented.

Pascal

```pascal
TVTButtonStyle = (bsRectangle, bsTriangle);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bsRectangle</td>
<td>traditional Windows look (plus/minus buttons)</td>
</tr>
<tr>
<td>bsTriangle</td>
<td>traditional Macintosh look</td>
</tr>
</tbody>
</table>

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTColumnOption Enumeration

Types

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

File

VirtualTrees

Links

Types
TVTDragImageKind Enumeration

Types

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>diComplete</td>
<td>show a complete drag image with all columns, only visible columns are shown</td>
</tr>
<tr>
<td>diMainColumnOnly</td>
<td>show only the main column (the tree column)</td>
</tr>
<tr>
<td>diNoImage</td>
<td>don't show a drag image at all</td>
</tr>
</tbody>
</table>

Group
Types

File
VirtualTrees

Links
Types
What do you think about this topic? Send feedback!
**TVTDragMoveRestriction Enumeration**

**Types**

Not documented.

**Pascal**

```
TVTDragMoveRestriction = (dmrNone, dmrHorizontalOnly);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
TVTDragType Enumeration

Types

Not documented.

Pascal

```pascal
TVTDragType = (dtOLE, dtVCL);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
**TVTDrawSelectionMode Enumeration**

**Types**
Not documented.

**Pascal**

```pascal
TVTDrawSelectionMode = (smDottedRectangle, smBlendedRectangle);
```

**Description**
Use other resources like the news group or the Delphi Gems message board to find a description.

**Members**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>smDottedRectangle</td>
<td>same as DrawFocusRect</td>
</tr>
<tr>
<td>smBlendedRectangle</td>
<td>alpha blending, uses special colors (see TVTColors)</td>
</tr>
</tbody>
</table>

**Group**
Types

**File**
VirtualTrees

**Links**
Types

*What do you think about this topic? Send feedback!*
TVTDropMarkMode Enumeration

**Types**

Not documented.

**Pascal**

```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

**File**

VirtualTrees

**Links**

Types

*What do you think about this topic? Send feedback!*
**TVTHeaderColumnLayout** Enumeration

**Types**

Not documented.

**Pascal**

```
TVTHeaderColumnLayout = (blGlyphLeft, blGlyphRight, ...
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? Send feedback!*
**TVTHeaderOption Enumeration**

**Types**

Not documented.

**Pascal**

```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**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hoAutoResize</td>
<td>adjust a column so that the header never exceeds client width of owner control</td>
</tr>
<tr>
<td>hoColumnResize</td>
<td>resizing columns is allowed</td>
</tr>
<tr>
<td>hoDblClickResize</td>
<td>allows a column to resize itself to its largest entry</td>
</tr>
<tr>
<td>hoDrag</td>
<td>dragging columns is allowed</td>
</tr>
<tr>
<td>hoHotTrack</td>
<td>header captions are highlighted when mouse is over a particular column</td>
</tr>
<tr>
<td>hoOwnerDraw</td>
<td>header items with the owner draw style can be drawn by the application via event</td>
</tr>
<tr>
<td><strong>hoRestrictDrag</strong></td>
<td>header can only be dragged horizontally</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td><strong>hoShowHint</strong></td>
<td>show application defined header hint</td>
</tr>
<tr>
<td><strong>hoShowImages</strong></td>
<td>show images</td>
</tr>
<tr>
<td><strong>hoShowSortGlyphs</strong></td>
<td>allow visible sort glyphs</td>
</tr>
<tr>
<td><strong>hoVisible</strong></td>
<td>header is visible</td>
</tr>
</tbody>
</table>

**Group**

**Types**

**File**

VirtualTrees

**Links**

**Types**

*What do you think about this topic? Send feedback!*
**TVTHeaderPopupOption Enumeration**

**Types**

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**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>poOriginalOrder</td>
<td>Show menu items in original column order as they were added to the tree.</td>
</tr>
<tr>
<td>poAllowHideAll</td>
<td>Allows to hide all columns, including the last one.</td>
</tr>
</tbody>
</table>

**Group**

Types

**File**

VTHeaderPopup

**Links**

Types
What do you think about this topic? Send feedback!
TVTMenuItem Type

Types

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

File

VTHHeaderPopup

Links

Types

What do you think about this topic? Send feedback!
**TVTHeaderStyle Enumeration**

**Types**

Not documented.

**Pascal**

```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**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsThickButtons</td>
<td>TButton look and feel</td>
</tr>
<tr>
<td>hsFlatButtons</td>
<td>flatter look than hsThickButton, like an always raised flat TToolButton</td>
</tr>
<tr>
<td>hsPlates</td>
<td>flat TToolButton look and feel (raise on hover etc.)</td>
</tr>
<tr>
<td>hsXPStyle</td>
<td>Windows XP style</td>
</tr>
</tbody>
</table>

**Group**

**Types**

**File**

VirtualTrees

**Links**
Types

What do you think about this topic? Send feedback!
TVTHintMode Enumeration

Types

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hmDefault</td>
<td>show the hint of the control</td>
</tr>
<tr>
<td>hmHint</td>
<td>show node specific hint string returned by the application</td>
</tr>
<tr>
<td>hmHintAndDefault</td>
<td>same as hmHint but show the control's hint if no node is concerned</td>
</tr>
<tr>
<td>hmTooltip</td>
<td>show the text of the node if it isn't already fully shown</td>
</tr>
</tbody>
</table>

Group

Types

File

VirtualTrees
What do you think about this topic? Send feedback!
**TVTImageInfoIndex Enumeration**

**Types**

Not documented.

**Pascal**

```pascal
TVTImageInfoIndex = (iiNormal, iiState, iiCheck, ii0
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

**Types**

**File**

VirtualTrees

**Links**

**Types**

---

*What do you think about this topic? Send feedback!*
**TVTImageKind Enumeration**

**Types**
Not documented.

**Pascal**

```pascal
TVTImageKind = (ikNormal, ikSelected, ikState, ikOve
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- Types

**File**

- VirtualTrees

**Links**

- Types

---

*What do you think about this topic? Send feedback!*
TVTIncrementalSearch Enumeration

Types

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isAll</td>
<td>search every node in tree, initialize if necessary</td>
</tr>
<tr>
<td>isNone</td>
<td>disable incremental search</td>
</tr>
<tr>
<td>isInitializedOnly</td>
<td>search only initialized nodes, skip others</td>
</tr>
<tr>
<td>isVisibleOnly</td>
<td>search only visible nodes, initialize if necessary</td>
</tr>
</tbody>
</table>
What do you think about this topic? Send feedback!
TVTInternalPaintOption Enumeration

Types

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>poBackground</td>
<td>draw background image if there is any and it is enabled</td>
</tr>
<tr>
<td>poColumnColor</td>
<td>erase node's background with the column's color</td>
</tr>
<tr>
<td>poDrawFocusRect</td>
<td>draw focus rectangle around the focused node</td>
</tr>
<tr>
<td>poDrawSelection</td>
<td>draw selected nodes with the normal selection color</td>
</tr>
<tr>
<td>poDrawDropMark</td>
<td>draw drop mark if a node is currently the drop target</td>
</tr>
<tr>
<td>poGridLines</td>
<td>draw grid lines if enabled</td>
</tr>
<tr>
<td>poMainOnly</td>
<td>draw only the main column</td>
</tr>
<tr>
<td>poSelectedOnly</td>
<td>draw only selected nodes</td>
</tr>
</tbody>
</table>
What do you think about this topic? Send feedback!
TVTLineMode Enumeration

Types

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lmNormal</td>
<td>usual tree lines (as in TTreeview)</td>
</tr>
<tr>
<td>lmBands</td>
<td>looks similar to a Nassi-Schneidermann diagram</td>
</tr>
</tbody>
</table>

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTLineStyle Enumeration

Types

Not documented.

Pascal

```pascal
TVTLineStyle = (lsCustomStyle, lsDotted, lsSolid);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Members

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsCustomStyle</td>
<td>application provides a line pattern</td>
</tr>
<tr>
<td>lsDotted</td>
<td>usual dotted lines (default)</td>
</tr>
<tr>
<td>lsSolid</td>
<td>simple solid lines</td>
</tr>
</tbody>
</table>

Group

Types

File

VirtualTrees

Links

Types
What do you think about this topic? Send feedback!
**TVTLineType Enumeration**

**Types**

Not documented.

**Pascal**

```
TVTLineType = (ltNone, ltBottomRight, ltTopDown, ltTopDownRight, ltRight, ltTopRight, ltLeft);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Members**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ltNone</td>
<td>no line at all</td>
</tr>
<tr>
<td>ltBottomRight</td>
<td>a line from bottom to the center and from there to the right</td>
</tr>
<tr>
<td>ltTopDown</td>
<td>a line from top to bottom</td>
</tr>
<tr>
<td>ltTopDownRight</td>
<td>a line from top to bottom and from center to the right</td>
</tr>
<tr>
<td>ltRight</td>
<td>a line from center to the right</td>
</tr>
<tr>
<td>ltTopRight</td>
<td>a line from bottom to center and from there to the right special styles for alternative drawings of tree lines</td>
</tr>
<tr>
<td>ltLeft</td>
<td>a line from top to bottom at the left</td>
</tr>
</tbody>
</table>
What do you think about this topic? Send feedback!
TVTMiscOption Enumeration

Types

Not documented.

Pascal

```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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>toAcceptOLEDrop</td>
<td>Register tree as OLE accepting drop target</td>
</tr>
<tr>
<td>toCheckSupport</td>
<td>Show checkboxes/radio buttons.</td>
</tr>
<tr>
<td>toEditable</td>
<td>Node captions can be edited.</td>
</tr>
<tr>
<td>toFullRepaintOnResize</td>
<td>Fully invalidate the tree when its window is resized (CS_HREDRAW/CS_VREDRAW).</td>
</tr>
<tr>
<td>toGridExtensions</td>
<td>Use some special enhancements to simulate and support grid behavior.</td>
</tr>
<tr>
<td>toInitOnSave</td>
<td>Initialize nodes when saving a tree to a stream.</td>
</tr>
<tr>
<td>toReportMode</td>
<td>Tree behaves like TListView in report mode.</td>
</tr>
<tr>
<td>Action</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>toToggleOnDb1Click</td>
<td>Toggle node expansion state when it is double clicked.</td>
</tr>
<tr>
<td>toWheelPanning</td>
<td>This option and toMiddleClickSelect are mutual exclusive, where panning has precedence.</td>
</tr>
<tr>
<td>toReadOnly</td>
<td>No action is executed and node editing is not possible.</td>
</tr>
</tbody>
</table>

**Group**

Types

**File**

VirtualTrees

**Links**

Types

What do you think about this topic? **Send feedback!**
TVTNodeAlignment Enumeration

Types

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>naFromBottom</td>
<td>the align member specifies amount of units (usually pixels) from top border of the node</td>
</tr>
<tr>
<td>naFromTop</td>
<td>align is to be measured from bottom</td>
</tr>
<tr>
<td>naProportional</td>
<td>align is to be measure in percent of the entire node height and relative to top</td>
</tr>
</tbody>
</table>

Group

Types

File

VirtualTrees
What do you think about this topic? Send feedback!
**TVTNodeAttachMode Enumeration**

**Types**

Not documented.

**Pascal**

```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**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>amNoWhere</td>
<td>just for simplified tests, means to ignore the Add/Insert command</td>
</tr>
<tr>
<td>amInsertBefore</td>
<td>insert node just before destination (as sibling of destination)</td>
</tr>
<tr>
<td>amInsertAfter</td>
<td>insert node just after destination (as sibling of destination)</td>
</tr>
<tr>
<td>amAddChildFirst</td>
<td>add node as first child of destination</td>
</tr>
<tr>
<td>amAddChildLast</td>
<td>add node as last child of destination</td>
</tr>
</tbody>
</table>

**Group**

**Types**
What do you think about this topic? Send feedback!
**TVTScrollbarShowEvent Type**

**Types**

Not documented.

**Pascal**

```
TVTScrollbarShowEvent = procedure (Sender: TBaseVirtualTree)
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Types

**File**

VirtualTrees

**Links**

Types

---

*What do you think about this topic? Send feedback!*
**TVTPaintOption Enumeration**

*Types*

Not documented.

*Pascal*

```pascal
TVTPaintOption = (toHideFocusRect, toHideSelection, ...
```

*Description*

Use other resources like the news group or the Delphi Gems message board to find a description.

*Members*

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>toHideFocusRect</td>
<td>Avoid drawing the dotted rectangle around the currently focused node.</td>
</tr>
<tr>
<td>toHideSelection</td>
<td>Selected nodes are drawn as unselected nodes if the tree is unfocused.</td>
</tr>
<tr>
<td>toHotTrack</td>
<td>Track which node is under the mouse cursor.</td>
</tr>
<tr>
<td>toPopupMode</td>
<td>Paint tree as would it always have the focus (useful for tree combo boxes etc.)</td>
</tr>
<tr>
<td>toShowBackground</td>
<td>Use the background image if there's one.</td>
</tr>
<tr>
<td></td>
<td>Display collapse/expand buttons</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>toShowButtons</td>
<td>left to a node.</td>
</tr>
<tr>
<td>toShowDropmark</td>
<td>Show the dropmark during drag'n drop operations.</td>
</tr>
<tr>
<td>toShowHorzGridLines</td>
<td>Display horizontal lines to simulate a grid.</td>
</tr>
<tr>
<td>toShowRoot</td>
<td>Show lines also at top level (does not show the hidden/internal root node).</td>
</tr>
<tr>
<td>toShowTreeLines</td>
<td>Display tree lines to show hierarchy of nodes.</td>
</tr>
<tr>
<td>toShowVertGridLines</td>
<td>Display vertical lines (depending on columns) to simulate a grid.</td>
</tr>
<tr>
<td>toThemeAware</td>
<td>Draw UI elements (header, tree buttons etc.) according to the current theme if enabled (Windows XP+ only, application must be themed).</td>
</tr>
<tr>
<td>toUseBlendedImages</td>
<td>Enable alpha blending for ghosted nodes or those which are being cut/copied.</td>
</tr>
<tr>
<td>toGhostedIfUnfocused</td>
<td>Ghosted images are still shown as ghosted if unfocused (otherwise the become non-ghosted images).</td>
</tr>
<tr>
<td>toFullVertGridLines</td>
<td>This option only has an effect if toShowVertGridLines is enabled too.</td>
</tr>
<tr>
<td>toAlwaysHideSelection</td>
<td>Do not draw node selection, regardless of focused state.</td>
</tr>
<tr>
<td>toUseBlendedSelection</td>
<td>Enable alpha blending for node selections.</td>
</tr>
</tbody>
</table>
Group
  Types

File
  VirtualTrees

Links
  Types

What do you think about this topic? Send feedback!
TVTSearchDirection Enumeration

Types

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

File

VirtualTrees

Links

Types

What do you think about this topic? Send feedback!
TVTSearchStart Enumeration

Types
Not documented.

Pascal

\[
\text{TVTSearchStart} = (\text{ssAlwaysStartOver, ssLastHit, ssFocusedNode})
\]

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Members

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssAlwaysStartOver</td>
<td>always use the first/last node (depending on direction) to search from</td>
</tr>
<tr>
<td>ssLastHit</td>
<td>use the last found node</td>
</tr>
<tr>
<td>ssFocusedNode</td>
<td>use the currently focused node</td>
</tr>
</tbody>
</table>

Group
Types

File
VirtualTrees

Links
Types
What do you think about this topic? Send feedback!
TVTSelectionOption Enumeration

Types

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

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>toDisableDrawSelection</td>
<td>Prevent user from selecting with the selection rectangle in multiselect mode.</td>
</tr>
<tr>
<td>toExtendedFocus</td>
<td>Entries other than in the main column can be selected, edited etc.</td>
</tr>
<tr>
<td>toFullRowSelect</td>
<td>Hit test as well as selection highlight are not constrained to the text of a node.</td>
</tr>
<tr>
<td>toLevelSelectConstraint</td>
<td>Constrain selection to the same level as the selection anchor.</td>
</tr>
<tr>
<td>toMiddleClickSelect</td>
<td>with the middle mouse button. This and toWheelPanning are mutual exclusive.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>toMultiSelect</td>
<td>Allow more than one node to be selected.</td>
</tr>
<tr>
<td>toRightClickSelect</td>
<td>with the right mouse button.</td>
</tr>
<tr>
<td>toSiblingSelectConstraint</td>
<td>constrain selection to nodes with same parent</td>
</tr>
<tr>
<td>toCenterScrollIntoView</td>
<td>Center nodes vertically in the client area when scrolling into view.</td>
</tr>
<tr>
<td>toSimpleDrawSelection</td>
<td>Simplifies draw selection, so a node's caption does not need to intersect with the selection rectangle.</td>
</tr>
</tbody>
</table>

**Group**
- Types

**File**
- VirtualTrees

**Links**
- Types

*What do you think about this topic? Send feedback!*
**TVTStringOption Enumeration**

**Types**

Not documented.

**Pascal**

```
TVTStringOption = (toSaveCaptions, toShowStaticText,
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Members**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>toSaveCaptions</td>
<td>If set then the caption is automatically saved with the tree node, regardless of what is saved in the user data.</td>
</tr>
<tr>
<td>toShowStaticText</td>
<td>Show static text in a caption which can be differently formatted than the caption but cannot be edited.</td>
</tr>
<tr>
<td>toAutoAcceptEditChange</td>
<td>If not set then changes are cancelled.</td>
</tr>
</tbody>
</table>

**Group**

Types

**File**
**TVTUpdateState Enumeration**

**Types**

Not documented.

**Pascal**

```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**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usBegin</td>
<td>The tree just entered the update state (BeginUpdate call for the first time).</td>
</tr>
<tr>
<td>usBeginSynch</td>
<td>The tree just entered the synch update state (BeginSynch call for the first time).</td>
</tr>
<tr>
<td>usSynch</td>
<td>Begin/EndSynch has been called but the tree did not change the update state.</td>
</tr>
<tr>
<td>usUpdate</td>
<td>Begin/EndUpdate has been called but the tree did not change the update state.</td>
</tr>
<tr>
<td>usEnd</td>
<td>The tree just left the update state (EndUpdate called for the last level).</td>
</tr>
</tbody>
</table>
The tree just left the synch update state (EndSynch called for the last level).

Group
   Types

File
   VirtualTrees

Links
   Types

What do you think about this topic? Send feedback!
Variables

These are all variables that are contained in this documentation.

Group
Virtual Treeview

Variables

- **CF_CSV**
  Not documented.
- **CF_HTML**
  Not documented.
- **CF_VIRTUALTREE**
  Not documented.
- **CF_VRTF**
  Not documented.
- **CF_VRTFNOOBBJS**
  Not documented.
- **CF_VTREFERENCE**
  Not documented.
- **ClipboardDescriptions**
  Not documented.
- **DarkCheckImages**
  Not documented.
- **DarkTickImages**
  Not documented.
- **FlatImages**
  Not documented.
- HintFont
  Not documented.
- HintWindowDestroyed
  Not documented.
- Initialized
  Not documented.
- InternalClipboardFormats
  Not documented.
- IsWin2K
  Not documented.
- IsWinNT
  Not documented.
- IsWinXP
  Not documented.
- LightCheckImages
  Not documented.
- LightTickImages
  Not documented.
- MMXAvailable
  Not documented.
- NeedToUninitialize
  Not documented.
- StandardOLEFormat
  Not documented.
- SystemCheckImages
  Not documented.
- SystemFlatCheckImages
  Not documented.
- UtilityImages
  Not documented.
- Watcher
  Not documented.
- WorkerThread
  Not documented.
- WorkEvent
Not documented.

XPImages
Not documented.

Legend

Variable

Links
Virtual Treeview, Variables, Legend

What do you think about this topic? Send feedback!
CF_CSV Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
CF_HTML Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
CFVIRTUALTREE Variable

Variables

Not documented.

Pascal

CFVIRTUALTREE: Word;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
CF_VRTF Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
CF_VRTFNOOBOJS Variable

Variables

Not documented.

Pascal

```pascal
CF_VRTFNOOBOJS: Word;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
CF_VTREFERENCE Variable

Variables

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

File
VirtualTrees

Links
Variables

What do you think about this topic? Send feedback!
ClipboardDescriptions Variable

Variables

Not documented.

Pascal

ClipboardDescriptions: array [1..CF_MAX - 1] of TClipboardFormatEntry

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
DarkCheckImages Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
DarkTickImages Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
FlatImages Variable

Variables

Not documented.

Pascal

\[
\text{FlatImages: TImageList;}
\]

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
HintFont Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
HintWindowDestroyed Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
Initialized Variable

Not documented.

Pascal

```pascal
Initialized: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
InternalClipboardFormats Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
IsWinNT Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
IsWinXP Variable

Variables

Not documented.

Pascal

```pascal
IsWinXP: Boolean;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
**LightCheckImages Variable**

Variables

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

**File**

VirtualTrees

**Links**

Variables

*What do you think about this topic? Send feedback!*
LightTickImages Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
**NeedToUninitialize Variable**

**Variables**

Not documented.

**Pascal**

```
NeedToUninitialize: Boolean;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Variables

**File**

VirtualTrees

**Links**

Variables

---

*What do you think about this topic?* Send feedback!
StandardOLEFormat Variable

Pascal

```pascal
StandardOLEFormat: TFormatEtc = ( cfFormat: 0; ptd: nil)
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
SystemCheckImages Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
**Watcher Variable**

Variables

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

**File**

VirtualTrees

**Links**

Variables

*What do you think about this topic? Send feedback!*
WorkerThread Variable

Variables

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

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
**WorkEvent Variable**

**Variables**

Not documented.

**Pascal**

```pascal
WorkEvent: THandle;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Variables

**File**

VirtualTrees

**Links**

Variables

---

*What do you think about this topic? Send feedback!*
XPIImages Variable

Variables

Not documented.

Pascal

XPIImages: TImageList;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Variables

File

VirtualTrees

Links

Variables

What do you think about this topic? Send feedback!
These are all constants that are contained in this documentation.

**Constants**

- **AlignmentToDrawFlag**
  Not documented.

- **AllocIncrement**
  Not documented.

- **BaseChunk**
  Not documented.

- **CacheThreshold**
  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**
  Not documented.

- **CFSTR_CSV**
  Contains the registration string for certain clipboard formats.

- **CFSTR_HTML**
  Contains the registration string for certain clipboard formats.

- **CFSTR_RTF**
  Contains the registration string for certain clipboard formats.

- **CFSTR_RTFNOOBS**
  Contains the registration string for certain clipboard formats.

- **CFSTR_VIRTUALTREE**
  Contains the registration string for certain clipboard formats.

- **CFSTR_VTREFERENCE**
Contains the registration string for certain clipboard formats.

- **ChangeTimer**
  Not documented.
- **ckButtonDisabled**
- **ckButtonHot**
- **ckButtonNormal**
- **ckButtonPressed**
- **ckCheckCheckedDisabled**
- **ckCheckCheckedHot**
- **ckCheckCheckedNormal**
- **ckCheckCheckedPressed**
- **ckCheckMixedDisabled**
- **ckCheckMixedHot**
- **ckCheckMixedNormal**
- **ckCheckMixedPressed**
- **ckCheckUncheckDisabled**
- **ckCheckUncheckHot**
- **ckCheckUncheckNormal**
- **ckCheckUncheckPressed**
- **ckEmpty**
- **ckRadioCheckedDisabled**
- **ckRadioCheckedHot**
- **ckRadioCheckedNormal**
- **ckRadioCheckedPressed**
- **ckRadioUncheckDisabled**
- **ckRadioUncheckHot**
- **ckRadioUncheckNormal**
- **ckRadioUncheckPressed**
- **ClipboardStates**
  Not documented.
- **CLSID_DragDropHelper**
  Not documented.
- **CM_AUTOADJUST**
Not documented.

CM_DENYSUBCLASSING
Not documented.

Copyright
Not documented.

crHeaderSplit
Not documented.

DefaultAnimationOptions
Not documented.

DefaultAutoOptions
Not documented.

DefaultColumnOptions
Not documented.

DefaultMiscOptions
Not documented.

DefaultPaintOptions
Not documented.

DefaultScrollUpdateFlags
Not documented.

DefaultSelectionOptions
Not documented.

DefaultStringOptions
Not documented.

EditTimer
Not documented.

ExpandTimer
Not documented.

FadeAnimationStepCount
Not documented.

Grays
Not documented.

hcTFCannotSetUserData
Not documented.

hcTFClipboardFailed
Not documented.
- `hcTFCorruptStream1`: Not documented.
- `hcTFCorruptStream2`: Not documented.
- `hcTFEditLinkIsNil`: Not documented.
- `hcTFStreamTooSmall`: Not documented.
- `hcTFWrongMoveError`: Not documented.
- `hcTFWrongStreamFormat`: Not documented.
- `hcTFWrongStreamVersion`: Not documented.
- `HeaderTimer`: Not documented.
- `IID_IDragSourceHelper`: Not documented.
- `IID_IDropTarget`: Not documented.
- `IID_IDropTargetHelper`: Not documented.
- `InvalidColumn`: Not documented.
- `MagicID`: Not documented.
- `MinimumTimerInterval`: Not documented.
- `MouseButtonDown`: Not documented.
- `NoColumn`: Not documented.
- `NodeChunk`: Not documented.
- `OptionMap`: Not documented.
Not documented.

- PressedState
  Not documented.
- RTLFlag
  Not documented.
- SCannotSetUserData
  Not documented.
- SClipboardFailed
  Not documented.
- SCorruptStream1
  Not documented.
- SCorruptStream2
  Not documented.
- ScrollTimer
  Not documented.
- SearchTimer
  Not documented.
- SEditLinkIsNil
  Not documented.
- ShadowSize
  Size in pixels of the hint shadow.
- SID_IDragSourceHelper
  Not documented.
- SID_IDropTarget
  Not documented.
- SID_IDropTargetHelper
  Not documented.
- SStreamTooSmall
  Not documented.
- StructureChangeTimer
  Not documented.
- SWrongMoveError
  Not documented.
- SWrongStreamFormat
  Not documented.
SWrongStreamVersion
Not documented.
SysGrays
Not documented.
TreeNodeSize
Not documented.
UnpressedState
Not documented.
UserChunk
Not documented.
UtilityImageSize
Not documented.
VTHeaderStreamVersion
Not documented.
VTStreamVersion
Not documented.
VTVersion
Not documented.
WideCR
Not documented.
WideLF
Not documented.
WideLineSeparator
Not documented.
WideNull
Not documented.
WM_CHANGETESTATE
Not documented.
XPDarkGradientColor
Not documented.
XPDarkSplitBarColor
Not documented.
XPDownInnerLineColor
Not documented.
XPDownMiddleLineColor
Not documented.

**XPDownOuterLineColor**
Not documented.

**XPLightSplitBarColor**
Not documented.

**XPMainHeaderColorDown**
Not documented.

**XPMainHeaderColorHover**
Not documented.

**XPMainHeaderColorUp**
Not documented.

**Group**

**Virtual Treeview**

**Topics**

Check button image indices

**Legend**

**Constant**

**Links**

Constants, Virtual Treeview, Topics, Legend

---

What do you think about this topic? Send feedback!
AlignmentToDrawFlag Constant

Constants

Not documented.

Pascal

AlignmentToDrawFlag: array[TAlignment] of Cardinal =

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
AllocIncrement Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
BaseChunk Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
CacheThreshold Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
CaptionChunk Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
CFSTR_CSV Constant

Contains the registration string for certain clipboard formats.

Pascal

```pascal
CFSTR_VIRTUALTREE = 'Virtual Tree Data';
CFSTR_VTREFERENCE = 'Virtual Tree Reference';
CFSTR_HTML = 'HTML Format';
CFSTR RTF = 'Rich Text Format';
CFSTR RTFNOOBS = '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.
- **RTFNOOBS**: 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 interception.

Group
  Constants

File
  VirtualTrees

Links
  Constants

What do you think about this topic? Send feedback!
**ChangeTimer Constant**

**Constants**

Not documented.

**Pascal**

```pascal
ChangeTimer = 5;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- Constants

**File**

- VirtualTrees

**Links**

- Constants

---

*What do you think about this topic? Send feedback!*
Check button image indices

Constants

Links

What do you think about this topic? Send feedback!
**ClipboardStates Constant**

**Constants**

Not documented.

**Pascal**

```pascal
ClipboardStates = [tsCopyPending, tsCutPending];
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- Constants

**File**

- VirtualTrees

**Links**

- Constants

---

*What do you think about this topic? Send feedback!*
**CLSID_DragDropHelper Constant**

**Constants**

Not documented.

**Pascal**

```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

**File**

- VirtualTrees

**Links**

- Constants

What do you think about this topic? Send feedback!
CM_AUTOADJUST Constant

Not documented.

Pascal

\[
\text{CM\_AUTOADJUST} = \text{CM\_BASE} + 2005;
\]

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
CM_DENYSUBCLASSING Constant

Not documented.

Pascal

```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

**File**

- VirtualTrees

**Links**

- Constants

*What do you think about this topic? Send feedback!*
Copyright Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
**crHeaderSplit Constant**

**Constants**

Not documented.

**Pascal**

```pascal
crHeaderSplit = TCursor(100);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

**Constants**

**File**

**VirtualTrees**

**Links**

**Constants**

---

*What do you think about this topic? [Send feedback]*
DefaultAnimationOptions Constant

Constants

Not documented.

Pascal

```pascal
DefaultAnimationOptions = [];```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
DefaultAutoOptions Constant

Not documented.

Pascal

```
DefaultAutoOptions = [toAutoDropExpand, toAutoTriState];
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
DefaultColumnOptions Constant

Constants

Not documented.

Pascal

```
DefaultColumnOptions = [coAllowClick, coDraggable, ...
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
DefaultMiscOptions Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
**DefaultPaintOptions Constant**

**Constants**

Not documented.

**Pascal**

```
DefaultPaintOptions = [toShowButtons, toShowButtons,
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

**Constants**

**File**

**VirtualTrees**

**Links**

**Constants**

---

*What do you think about this topic? Send feedback!*
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

File

 VirtualTrees

Links

 Constants

What do you think about this topic? Send feedback!
DefaultSelectionOptions Constant

Constants

Not documented.

Pascal

```
DefaultSelectionOptions = [];  
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
DefaultStringOptions Constant

Not documented.

Pascal

```
DefaultStringOptions = [toSaveCaptions, toAutoAccept]
```

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Group
 Constants

File
 VirtualTrees

Links
 Constants

What do you think about this topic? Send feedback!
EditTimer Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
FadeAnimationStepCount Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
Grays Constant

Constants

Not documented.

Pascal

```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

File

VirtualTrees

Links

Constants

---

What do you think about this topic? Send feedback!
hcTFCannotSetUserData Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File
VirtualTrees

Links
Constants

What do you think about this topic? Send feedback!
hcTFCorruptStream1 Constant

Constants

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

File

VirtualTrees

Links

Constants
hcTFCorruptStream2 Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
hcTFEditLinkIsNil Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
hcTFStreamTooSmall Constant

**Constants**

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

**File**

VirtualTrees

**Links**

Constants

*What do you think about this topic? Send feedback!*
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
hcTFWrongStreamVersion Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
HeaderTimer Constant

Not documented.

Pascal

```pascal
HeaderTimer = 3;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
IID_IDropTarget Constant

Constants

Not documented.

Pascal

```
IID_IDropTarget: TGUID = (D1: $00000122; D2: $0000);
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
IID_IDropTargetHelper Constant

Constants

Not documented.

Pascal

```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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
InvalidColumn Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
MagicID Constant

Not documented.

Pascal

```pascal
MagicID: TMagicID = (#$2045, 'V', 'T', WideChar(VTTreeVersion))
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
MinimumTimerInterval Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
**MouseButtonDown Constant**

**Constants**

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

**File**

- VirtualTrees

**Links**

- Constants

*What do you think about this topic? Send feedback!*
NoColumn Constant

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Pascal

```pascal
NoColumn = -1;
```

Group
Constants

File
VirtualTrees

Links
Constants

What do you think about this topic? Send feedback!
### NodeChunk Constant

**Constants**

Not documented.

**Pascal**

```pascal
NodeChunk = 1;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- Constants

**File**

- VirtualTrees

**Links**

- Constants

---

*What do you think about this topic? Send feedback!*
OptionMap Constant

Constants

Not documented.

Pascal

OptionMap: array[TOldVTOption] of Integer = ( Ord(toAcceptOLEDrop), Ord(toAnimatedToggle), Ord(toAutoDropExpand), Ord(toAutoExpand), Ord(toAutoScroll), Ord(toAutoSelect), Ord(toAutoSelectOnDblClick), Ord(toAutoSelectOnDown), Ord(toAutoSelectOnUp), Ord(toAutoSelectTo), Ord(toShowFields), 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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
**PressedState Constant**

**Constants**

Not documented.

**Pascal**

```pascal
PressedState: array[TCheckState] of TCheckState = (  
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- Constants

**File**

- VirtualTrees

**Links**

- Constants

---

*What do you think about this topic? Send feedback!*
**RTLFlag Constant**

Constants

Not documented.

**Pascal**

\[
\text{RTLFlag: array[Boolean] of Integer} = (0, \text{ETO_RTLREAD})
\]

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Constants

**File**

VirtualTrees

**Links**

Constants

*What do you think about this topic? Send feedback!*
**SCannotSetUserData Constant**

**Constants**

Not documented.

**Pascal**

```pascal
SCannotSetUserData = 'Cannot set initial user data because there is not enough user data allocated.'
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Constants

**File**

VirtualTrees

**Links**

Constants

*What do you think about this topic? Send feedback!*
SClipboardFailed Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
SCorruptStream1 Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
SCorruptStream2 Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
ScrollTimer Constant

**Constants**

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

**File**

- VirtualTrees

**Links**

- Constants

*What do you think about this topic? Send feedback!*
SearchTimer Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
SEditLinkIsNil Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
ShadowSize Constant

**Constants**

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

**File**

VirtualTrees

**Links**

Constants

*What do you think about this topic? Send feedback!*
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
**SID_IDropTarget Constant**

**Constants**

<table>
<thead>
<tr>
<th>SID_IDragSourceHelper</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID_IDropTargetHelper</td>
<td>Constant</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SID_IDropTarget</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>'{00000122-0000-0000-C000-000000000046}'</td>
<td></td>
</tr>
</tbody>
</table>

**Pascal**

```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

**File**

- VirtualTrees

**Links**

- Constants

---

*What do you think about this topic? Send feedback!*

---
**SID_IDropTargetHelper Constant**

**Constants**

Not documented.

**Pascal**

```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

**File**

- VirtualTrees

**Links**

- Constants

*What do you think about this topic? Send feedback!*
SStreamTooSmall Constant

Constants

Not documented.

Pascal

```pascal
SStreamTooSmall = 'Unable to load tree structure, not
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
SWrongMoveError Constant

Not documented.

Pascal

```pascal
SWrongMoveError = 'Target node cannot be a child node.'
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

- Constants

File

- VirtualTrees

Links

- Constants

What do you think about this topic? Send feedback!
**SWrongStreamFormat Constant**

**Constants**

Not documented.

**Pascal**

```pascal
SWrongStreamFormat = 'Unable to load tree structure,
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

- Constants

**File**

- VirtualTrees

**Links**

- Constants

---

*What do you think about this topic? Send feedback!*
SWrongStreamVersion Constant

Constants

Not documented.

Pascal

```pascal
SWrongStreamVersion = 'Unable to load tree structure'
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
SysGrays Constant

**Constants**

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

**File**

VirtualTrees

**Links**

Constants

_What do you think about this topic? Send feedback!_
TreeNodeSize Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
UnpressedState Constant

Constants

Not documented.

Pascal

UnpressedState: array[TCheckState] of TCheckState =

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
UserChunk Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
UtilityImageSize Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
**VTHeaderStreamVersion Constant**

**Constants**

- 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**

**File**

VirtualTrees

**Links**

**Constants**

---

*What do you think about this topic? Send feedback!*
VTTreeStreamVersion Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
WideCR Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File
VirtualTrees

Links
Constants

What do you think about this topic? Send feedback!
**WideNull Constant**

**Constants**

Not documented.

**Pascal**

```pascal
WideNull = WideChar(#0);
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

Constants

**File**

VirtualTrees

**Links**

Constants

*What do you think about this topic? Send feedback!*
WM_CHANGESTATE Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
**XPDarkGradientColor Constant**

**Constants**

Not documented.

**Pascal**

```pascal
XPDarkGradientColor = $B8C7CB;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Group**

**Constants**

**File**

VirtualTrees

**Links**

**Constants**

---

*What do you think about this topic? Send feedback!*
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
XPDownMiddleLineColor Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
XPLightSplitBarColor Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
XPMainHeaderColorDown Constant

Constants

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

File

VirtualTrees

Links

Constants

What do you think about this topic? Send feedback!
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

File
VirtualTrees

Links
Constants

What do you think about this topic? Send feedback!
XPMainHeaderColorUp Constant

 Constants

 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

 File

 VirtualTrees

 Links

 Constants

 What do you think about this topic? Send feedback!
Symbol Reference

These are all symbols available in this documentation.

Group
Virtual Treeview

Interfaces

- IDragSourceHelper
  Not documented.
- IDropTargetHelper
  Not documented.
- IVTDragManager
  Not documented.
- IVTEditLink
  Interface which is used for communication between the treeview and a node editor.

Legend

- Class

Links
Virtual Treeview, Interfaces, Legend

What do you think about this topic? Send feedback!
IDragSourceHelper Interface

Symbol Reference | Methods | Legend

Not documented.

Pascal

```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

Members

Methods

- InitializeFromBitmap
  Not documented.
- InitializeFromWindow
  Not documented.

Legend

- public
- Method

Class Hierarchy

```
IUnknown  ───> IDragSourceHelper
```
File
VirtualTrees

Links
Symbol Reference, Methods, Legend

What do you think about this topic? Send feedback!
**IDragSourceHelper.InitializeFromBitmap Method**

*IDragSourceHelper Interface*

Not documented.

**Pascal**

```pascal
[SID_IDragSourceHelper]
function InitializeFromBitmap(var SHDragImage: TSHDragImage): integer;
```

**Description**

Use other resources like the news group or the Delphi Gems message board to find a description.

**Interface**

*IDragSourceHelper Interface*

**Links**

*IDragSourceHelper Interface*

*What do you think about this topic? Send feedback!*
IDragSourceHelper.InitializeFromWindow Method

IDragSourceHelper Interface

Not documented.

Pascal

```pascal
function InitializeFromWindow(Window: HWND; var ppt: TPoint; var pDataObject: IDataObject): HRESULT;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IDragSourceHelper Interface

Links

IDragSourceHelper Interface

What do you think about this topic? Send feedback!
IDropTargetHelper Interface

Description
Use other resources like the news group or the Delphi Gems message board to find a description.

Group
Symbol Reference

Members

Methods

- DragEnter
  Not documented.
- DragLeave
  Not documented.
- DragOver
  Not documented.
- Drop
  Not documented.
- Show
  Not documented.

Legend
What do you think about this topic? Send feedback!
IDropTargetHelper.DragEnter Method
IDropTargetHelper Interface

Not documented.

Pascal

```pascal
function DragEnter(hwndTarget: HWND; pDataObject: IDataObject; Description: WideString): HRESULT;
```

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IDropTargetHelper Interface

Links

IDropTargetHelper Interface

What do you think about this topic? Send feedback!
**IDropTargetHelper.DragLeave Method**

**IDropTargetHelper Interface**

Not documented.

**Pascal**

```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

**Links**

IDropTargetHelper Interface

*What do you think about this topic? Send feedback!*
IDropTargetHelper.DragOver Method

Not documented.

Pascal

```pascal
function DragOver(var ppt: TPoint; dwEffect: Integer): HRESULT;
```

Description

Use other resources like the newsgroup or the Delphi Gems message board to find a description.

Interface

IDropTargetHelper Interface

Links

IDropTargetHelper Interface

What do you think about this topic? Send feedback!
IDropTargetHelper.Drop Method
IDropTargetHelper Interface

Not documented.

Pascal

function Drop(pDataObject: IDataObject; var ppt: TPoint; dwEffect: Integer): HRESULT;

Description

Use other resources like the news group or the Delphi Gems message board to find a description.

Interface

IDropTargetHelper Interface

Links

IDropTargetHelper Interface

What do you think about this topic? Send feedback!
IDropTargetHelper.Show Method
IDropTargetHelper Interface

Not documented.

Pascal

```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

Links

IDropTargetHelper Interface

What do you think about this topic? Send feedback!
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

Members

Properties

DataObject
Not documented.

DragSource
Not documented.

DropTargetHelperSupported
Not documented.

IsDropTarget
Not documented.

Methods

ForceDragLeave
Not documented.
GetDataObject
Not documented.
GetDragSource
Not documented.
GetDropTargetHelperSupported
Not documented.
GetIsDropTarget
Not documented.

Legend

public
Property
read only
Method

Class Hierarchy

File
VirtualTrees

Links
Symbol Reference, Methods, Properties, Legend

What do you think about this topic? Send feedback!
IVTDragManager.DataObject Property

IVTDragManager Interface

Not documented.

Pascal

```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

Links

IVTDragManager Interface

---

What do you think about this topic? Send feedback!
IVTDragManager.DragSource Property
IVTDragManager Interface

Not documented.

Pascal

```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

Links
IVTDragManager Interface

What do you think about this topic? Send feedback!
**IVTDragManager.DropTargetHelperSupported Property**

**IVTDragManager Interface**

Not documented.

**Pascal**

```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

**Links**

IVTDragManager Interface

---

*What do you think about this topic? Send feedback!*
IVTDragManager.IsDropTarget Property

IVTDragManager Interface

Not documented.

Pascal

```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

Links

IVTDragManager Interface

What do you think about this topic? Send feedback!
IVTDragManager.ForceDragLeave Method

IVTDragManager Interface

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

Links

IVTDragManager Interface

What do you think about this topic? Send feedback!
IVTDragManager.GetDataObject Method

IVTDragManager Interface

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

Links

IVTDragManager Interface

What do you think about this topic? Send feedback!
IVTDragManager.GetDragSource Method

IVTDragManager Interface

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

Links

IVTDragManager Interface

What do you think about this topic? Send feedback!
IVTDragManager.GetDropTargetHelperSupported Method

IVTDragManager Interface

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

Links

IVTDragManager Interface

What do you think about this topic? Send feedback!
IVTDragManager.GetIsDropTarget Method

IVTDragManager Interface

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

Links

IVTDragManager Interface

What do you think about this topic? Send feedback!
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 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**

**Members**

**Methods**

- **BeginEdit**
  This function will be called by the virtual tree when the editing starts.

- **CancelEdit**
  This function will be called by the virtual tree when the current editing is about to be cancelled.

- **EndEdit**
  This function will be called by the virtual tree when the current editing is being finished.

- **GetBounds**
  The virtual tree can use this function to get the current bounding rect of the node editor.

- **PrepareEdit**
  This function is called by a virtual tree to initialize the node editor.

- **ProcessMessage**
  This function is used to forward messages being directed to the virtual tree.

- **SetBounds**
  The virtual tree calls this function to initialize the bounding rectangle of the node editor.

**Legend**

- public
Method

Class Hierarchy

File

VirtualTrees

Links

Symbol Reference, Methods, Legend

What do you think about this topic? Send feedback!
**IVTEditLink.BeginEdit Method**

**IVTEditLink Interface**

This function will be called by the virtual tree when the editing starts.

**Pascal**

```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` and `SetBounds` are executed.

**Interface**

IVTEditLink Interface

**Links**

IVTEditLink Interface

What do you think about this topic? Send feedback!
IVTEditLink.CancelEdit Method

This function will be called by the virtual tree when the current editing is about to be cancelled.

Pascal

```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

Links
  IVTEditLink Interface

What do you think about this topic? Send feedback!
IVTEditLink.EndEdit Method

IVTEditLink Interface

This function will be called by the virtual tree when the current editing is being finished.

Pascal

```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:

```pascal
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;
end;
```

**Interface**

IVTEditLink Interface

**Links**

IVTEditLink Interface

What do you think about this topic? Send feedback!
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.
**IVTEditLink.PrepareEdit Method**

**IVTEditLink Interface**

This function is called by a virtual tree to initialize the node editor.

**Pascal**

```pascal
function PrepareEdit(Tree: TBaseVirtualTree; Node: PVirtualNode) ;
```

**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** may be called anytime after this function returns. If the initialization fails simply return false (exceptions should be trapped).

**Interface**

**IVTEditLink Interface**

**Links**

**IVTEditLink Interface**

---

*What do you think about this topic? Send feedback!*
IVTEditLink.ProcessMessage Method

This function is used to forward messages being directed to the virtual tree.

Pascal

```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

Links

IVTEditLink Interface

What do you think about this topic? Send feedback!
**IVTEditLink.SetBounds Method**

**IVTEditLink Interface**

The virtual tree calls this function to initialize the bounding rectangle of the node editor.

**Pascal**

```pascal
procedure SetBounds(R: TRect); stdcall;
```

**Description**

This function is usually called after **PrepareEdit** and before **BeginEdit** 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:

```pascal
procedure EditLinkSetBounds(R: TRect); stdcall;
procedure IVTEditLink.SetBounds = EditLinkSetBounds
```
Interface
IVTEditLink Interface

Links
IVTEditLink Interface

What do you think about this topic? Send feedback!
Classes

Classes

- **EVirtualTreeError**
- **TBaseVirtualTree**
  TBaseVirtualTree is the main and base class for all other Virtual Treeview descendants.
- **TBufferedString**
- **TClipboardFormatList**
  Not documented.
- **TClipboardFormats**
  List of strings describing clipboard formats.

- **TCriticalSection**
  Not documented.
- **TCustomStringTreeOptions**
  Enhanced options class for string trees.
- **TCustomVirtualDrawTree**
  Simple owner draw descendant of the base tree.
- **TCustomVirtualStringTree**
  Descendant of **TBaseVirtualTree**, which is able to manage node captions on its own.
- **TCustomVirtualTreeOptions**
  Organizes all tree options into subproperties for easier management.
- **TEnumFormatEtc**
- **TScrollBarOptions**
- **TStringEditLink**
  TStringEditLink is the standard node editor of a **TVirtualStringTree**.
- **TStringTreeOptions**
  Options class used in the string tree and its descendents.
TVirtualDrawTree
Descendant of TBaseVirtualTree, which passes node paint events through to the application (similar to a draw grid)

TVirtualStringTree
Descendant of TBaseVirtualTree which is able to manage node captions on its own.

TVirtualTreeColumn
Represents a column in a Virtual Treeview.

TVirtualTreeColumns
Collection class, which holds the columns for the tree.

TVirtualTreeHintWindow
Internally used hint window class to support Unicode hints.

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

TVTColors
Collects all color related options for the tree control.

TVTDataObject
Implementation of an IDataObject interface.

TVTDragImage
Not documented.

TVTDragManager
Not documented.

TVTEdit
Not documented.

TVTHeader
Not documented.

TVTHeaderPopupMenu
Not documented.

TWideBufferedString
Not documented.

TWorkerThread
Not documented.

TWriterHack
Not documented.
TBaseVirtualTree Class Events

TBaseVirtualTree Class | Legend

Events

- **OnAdvancedHeaderDraw**
  Header paint support event.

- **OnAfterCellPaint**
  Paint support event.

- **OnAfterItemErase**
  Paint support event.

- **OnAfterItemPaint**
  Paint support event.

- **OnAfterPaint**
  Paint support event.

- **OnBeforeCellPaint**
  Paint support event.

- **OnBeforeItemErase**
  Paint support event.

- **OnBeforeItemPaint**
  Paint support event.

- **OnBeforePaint**
  Paint support event.

- **OnChange**
  Navigation support event.

- **OnChecked**
  Check support event.

- **OnChecking**
  Check support event.

- **OnCollapsed**
  Miscellaneous event.
OnCollapsing
  Miscellaneous event.
OnColumnClick
  Header and column support event.
OnColumnDbClick
  Header and column support event.
OnColumnResize
  Header and column support routine.
OnCompareNodes
  Sort and search support event.
OnCreateDataObject
  Drag'n drop support event.
OnCreateDragManager
  Drag'n drop support event.
OnCreateEditor
  Editing support event.
OnDragAllowed
  Drag'n drop support event.
OnDragDrop
  Drag'n drop support event.
OnDragOver
  Drag'n drop support event.
OnEditCancelled
  Editing support event.
OnEdited
  Editing support event.
OnEditing
  Editing support event.
OnExpanded
  Miscellaneous event.
OnExpanding
  Miscellaneous event.
OnFocusChanged
  Navigation support event.
OnFocusChanging
Navigation support event.

- **OnFreeNode**
  Data management node.

- **OnGetCellsEmpty**
  Triggered when the tree control needs to know whether a given column is empty.

- **OnGetCursor**
  Miscellaneous event.

- **OnGetHeaderCursor**
  Header and column support event.

- **OnGetHelpContext**
  Miscellaneous event.

- **OnGetImageIndex**
  Display management event.

- **OnGetImageIndexEx**
  Not documented.

- **OnGetLineStyle**
  Display management event.

- **OnGetNodeDataSize**
  Data management event.

- **OnGetPopupMenu**
  Miscellaneous event.

- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.

- **OnHeaderClick**
  Header & column support event.

- **OnHeaderDbClick**
  Header & column support event.

- **OnHeaderDragged**
  Header & column support event.

- **OnHeaderDraggedOut**
  Header & column support event.

- **OnHeaderDragging**
  Header & column support event.

- **OnHeaderDraw**
Header & column support event.
OnHeaderDrawQueryElements
Header & column support event.
OnHeaderMouseDown
Header & column support event.
OnHeaderMouseMove
Header & column support event.
OnHeaderMouseUp
Header & column support event.
OnHotChange
Navigation support event.
OnIncrementalSearch
Miscellaneous event.
OnInitChildren
Node management event.
OnInitNode
Node management event.
OnKeyAction
Miscellaneous event.
OnLoadNode
Streaming support event.
OnMeasureItem
Miscellaneous event.
OnNodeCopied
Miscellaneous event.
OnNodeCopying
Miscellaneous event.
OnNodeMoved
Miscellaneous event.
OnNodeMoving
Miscellaneous event.
OnPaintBackground
Paint support event.
OnRenderOLEData
Drag'n drop and clipboard support event.
- **OnResetNode**
  Node management event.
- **OnSaveNode**
  Streaming support event.
- **OnScroll**
  Miscellaneous event.
- **OnShowScrollbar**
  Not documented.
- **OnStateChange**
  Miscellaneous event.
- **OnStructureChange**
  Miscellaneous event.
- **OnUpdating**
  Miscellaneous event.
TBaseVirtualTree Class Methods

Methods

- **AbsoluteIndex**
  Reads the overall index of a node.

- **AddChild**
  Creates and adds a new child node to given node.

- **AddFromStream**
  Adds the content from the given stream to the given node.

- **AddToSelection**
  Adds one or more nodes to the current selection.

- **AdjustPaintCellRect**
  Used in descendents to modify the clip rectangle of the current column while painting a certain node.

- **AdjustPanningCursor**
  Loads the proper cursor which indicates into which direction scrolling is done.

- **AdviseChangeEvent**
  Used to register a delayed change event.

- **AllocateInternalDataArea**
  Registration method to allocate tree internal data per node.

- **Animate**
  Support method for animated actions in the tree view.

- **Assign**
  Used to copy properties from another Virtual Treeview.

- **BeginDrag**
  Starts an OLE drag'n drop operation.

- **BeginSynch**
  Enters the tree into a special synchronized mode.

- **BeginUpdate**
Locks the tree view to perform several update operations.

**CalculateSelectionRect**
Support method for draw selection.

**CanAutoScroll**
Determines whether the tree can currently auto scroll its window.

**CancelCutOrCopy**
Cancels any pending cut or copy clipboard operation.

**CancelEditNode**
Canceled the current edit operation, if there is any.

**CanEdit**
Determines whether a node can be edited or not.

**CanFocus**
Support method to determine whether the tree window can receive the input focus.

**CanShowDragImage**
Determines whether a drag image should be shown.

**Change**
Central method called when a node’s selection state changes.

**ChangeScale**
Helper method called by the VCL when control resizing is due.

**CheckParentCheckState**
Helper method for recursive check state changes.

**Clear**
Cleans the tree and removes all nodes.

**ClearChecked**
Not documented.

**ClearSelection**
Removes all nodes from the current selection.

**ClearTempCache**
Helper method to clear the internal temporary node cache.

**ColumnIsEmpty**
Used to determine if a cell is considered as being empty.

**CopyTo**
Copies **Source** and all its child nodes to **Target**.

**CopyToClipBoard**
Copies all currently selected nodes to the clipboard.

- **CountLevelDifference**
  Determines the level difference of two nodes.

- **CountVisibleChildren**
  Determines the number of visible child nodes of the given node.

- **Create**
  Constructor of the control

- **CreateParams**
  Prepares the creation of the controls window handle.

- **CreateWnd**
  Initializes data, which depends on the window handle.

- **CutToClipBoard**
  Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

- **DefineProperties**
  Helper method to customize loading and saving persistent tree data.

- **DeleteChildren**
  Removes all child nodes from the given node.

- **DeleteNode**
  Removes the given node from the tree.

- **DeleteSelectedNodes**
  Removes all currently selected nodes form the tree.

- **Destroy**
  Destructor of the control.

- **DetermineHiddenChildrenFlag**
  Determines whether all children of a given node are hidden.

- **DetermineHiddenChildrenFlagAllNodes**
  Determines whether all children of all nodes are hidden.

- **DetermineHitPositionLTR**
  Determines the hit position within a node with left-to-right and right-to-left orientation.

- **DetermineHitPositionRTL**
  Determines the hit position within a node with left-to-right and right-to-left orientation.

- **DetermineNextCheckState**
Not documented.

- **DetermineScrollDirections**
  - Not documented.

- **DoAdvancedHeaderDraw**
  - Not documented.

- **DoAfterCellPaint**
  - Not documented.

- **DoAfterItemErase**
  - Not documented.

- **DoAfterItemPaint**
  - Not documented.

- **DoAfterPaint**
  - Not documented.

- **DoAutoScroll**
  - Enables or disables the auto scroll timer.

- **DoBeforeCellPaint**
  - Not documented.

- **DoBeforeDrag**
  - Not documented.

- **DoBeforeItemErase**
  - Not documented.

- **DoBeforeItemPaint**
  - Not documented.

- **DoBeforePaint**
  - Not documented.

- **DoCancelEdit**
  - Called when the tree should stop editing without accepting changed values.

- **DoCanEdit**
  - Not documented.

- **DoChange**
  - Not documented.

- **DoCheckClick**
  - Not documented.

- **DoChecked**
Not documented.

- **DoChecking**
  Not documented.

- **Docollapsed**
  Not documented.

- **DoCollapsing**
  Not documented.

- **DoColumnClick**
  Not documented.

- **DoColumnDblClick**
  Not documented.

- **DoColumnResize**
  Not documented.

- **DoCompare**
  Not documented.

- **DoCreateDataObject**
  Not documented.

- **DoCreateDragManager**
  Not documented.

- **DoCreateEditor**
  Not documented.

- **DoDragDrop**
  Not documented.

- **DoDragExpand**
  Not documented.

- **DoDragging**
  Internal method which handles drag' drop.

- **DoDragOver**
  Not documented.

- **DoEdit**
  Initiates editing of the currently set focused column and edit node.

- **DoEndDrag**
  Not documented.

- **DoEndEdit**
  Stops the current edit operation and takes over the new content.
- **DoExpanded**
  Not documented.
- **DoExpanding**
  Not documented.
- **DoFocusChange**
  Not documented.
- **DoFocusChanging**
  Not documented.
- **DoFocusNode**
  Internal method to set the focused node.
- **DoFreeNode**
  Not documented.
- **DoGetAnimationType**
  Determines the type of animation to be used.
- **DoGetCursor**
  Not documented.
- **DoGetHeaderCursor**
  Not documented.
- **DoGetImageIndex**
  Not documented.
- **DoGetLineStyle**
  Not documented.
- **DoGetNodeHint**
  Not documented.
- **DoGetNodeTooltip**
  Not documented.
- **DoGetNodeWidth**
  Overridable method which always returns 0.
- **DoGetPopupMenu**
  Overridable method which triggers the OnGetPopup event.
- **DoGetUserClipboardFormats**
  Not documented.
- **DoHeaderClick**
  Not documented.
- **DoHeaderDblClick**
Not documented.

DoHeaderDragged
Not documented.

DoHeaderDraggedOut
Not documented.

DoHeaderDragging
Not documented.

DoHeaderDraw
Not documented.

DoHeaderDrawQueryElements
Not documented.

DoHeaderMouseDown
Not documented.

DoHeaderMouseMove
Not documented.

DoHeaderMouseUp
Not documented.

DoHotChange
Not documented.

DoIncrementalSearch
Not documented.

DoInitChildren
Not documented.

DoInitNode
Not documented.

DoKeyAction
Not documented.

DoLoadUserData
Not documented.

DoMeasureItem
Not documented.

DoNodeCopied
Not documented.

DoNodeCopying
Not documented.
DoNodeMoved
Not documented.

DoNodeMoving
Not documented.

DoPaintBackground
Not documented.

DoPaintDropMark
Overridable method which draws the small line on top of a nodes image depending on the current drop state.

DoPaintNode
Overridable method which does nothing.

DoPopupMenu
Overridable method which shows the popup menu for the given node.

DoRenderOLEData
Not documented.

DoReset
Not documented.

DoSaveUserData
Not documented.

DoScroll
Overridable method which triggers the OnScroll event.

DoSetOffsetXY
Internal core routine to set the tree's scroll position.

DoShowScrollbar
Not documented.

DoStartDrag
Not documented.

DoStateChange
Not documented.

DoStructureChange
Not documented.

DoTimerScroll
Callback method which is triggered whenever the scroll timer fires.

DoUpdating
Not documented.
**DoValidateCache**
Not documented.

**DragCanceled**
Called by the VCL when a drag'n drop operation was canceled by the user.

**DragDrop**
Helper method, which is used when a drag operation is finished.

**DragEnter**
Not documented.

**DragFinished**
Called when a drag operation is finished (accepted or cancelled).

**Dragging**
Returns true if a drag'n drop operation is in progress.

**DragLeave**
Not documented.

**DragOver**
Not documented.

**DrawDottedHLine**
Not documented.

**DrawDottedVLine**
Not documented.

**EditNode**
Starts editing the given node if allowed to.

**EndEditNode**
Stops node editing if it was started before.

**EndSynch**
Counterpart to **BeginSynch**.

**EndUpdate**
Resets the update lock set by **BeginUpdate**.

**ExecuteAction**
Not documented.

**FindNodeInSelection**
Helper method to find the given node in the current selection.

**FinishChunkHeader**
Not documented.
FinishCutOrCopy
Stops any pending cut or copy clipboard operation.

FlushClipboard
Renders all pending clipboard data.

FontChanged
Not documented.

FullCollapse
Collapses all nodes in the tree.

FullExpand
Expands all nodes in the tree.

GetBorderDimensions
Not documented.

GetCheckImage
Not documented.

GetCheckImageListFor
Not documented.

GetColumnClass
Returns the class to be used to manage columns in the tree.

GetControlsAlignment
Not documented.

GetDisplayRect
Returns the visible region used by the given node in client coordinates.

GetFirst
Group of node navigation functions.

GetFirstChecked
Not documented.

GetFirstChild
Group of node navigation functions.

GetFirstCutCopy
Group of node navigation functions.

GetFirstInitialized
Group of node navigation functions.

GetFirstNoInit
Group of node navigation functions.
GetFirstSelected
Group of node navigation functions.

GetFirstVisible
Group of node navigation functions.

GetFirstVisibleChild
Group of node navigation functions.

GetFirstVisibleChildNoInit
Group of node navigation functions.

GetFirstVisibleNoInit
Group of node navigation functions.

GetHeaderClass
Returns the header class to be used by the tree.

GetHintWindowClass
Not documented.

GetHitTestInfoAt
Returns information about the node at the given position.

GetImageIndex
Not documented.

GetLast
Group of node navigation functions.

GetLastChild
Group of node navigation functions.

GetLastChildNoInit
Group of node navigation functions.

GetLastInitialized
Group of node navigation functions.

GetLastNoInit
Group of node navigation functions.

GetLastVisible
Group of node navigation functions.

GetLastVisibleChild
Group of node navigation functions.

GetLastVisibleChildNoInit
Group of node navigation functions.

GetLastVisibleNoInit
Group of node navigation functions.

GetMaxColumnWidth
Returns the width of the largest node in the given column.

GetMaxRightExtend
Determines the maximum width of the currently visible part of the tree.

GetNativeClipboardFormats
Used to let descendants and the application add their own supported clipboard formats.

GetNext
Group of node navigation functions.

GetNextChecked
Not documented.

GetNextCutCopy
Group of node navigation functions.

GetNextInitialized
Group of node navigation functions.

GetNextNoInit
Group of node navigation functions.

GetNextSelected
Group of node navigation functions.

GetNextSibling
Group of node navigation functions.

GetNextVisible
Group of node navigation functions.

GetNextVisibleNoInit
Group of node navigation functions.

GetNextVisibleSibling
Group of node navigation functions.

GetNextVisibleSiblingNoInit
Group of node navigation functions.

GetNodeAt
Not documented.

GetNodeData
Returns the address of the user data area of the given node.

GetNodeLevel
Returns the indentation level of the given node.

- **GetOptionsClass**
  Customization helper to determine which options class the tree should use.

- **GetPrevious**
  Group of node navigation functions.

- **GetPreviousInitialized**
  Group of node navigation functions.

- **GetPreviousNoInit**
  Group of node navigation functions.

- **GetPreviousSibling**
  Group of node navigation functions.

- **GetPreviousVisible**
  Group of node navigation functions.

- **GetPreviousVisibleNoInit**
  Group of node navigation functions.

- **GetPreviousVisibleSibling**
  Group of node navigation functions.

- **GetPreviousVisibleSiblingNoInit**
  Group of node navigation functions.

- **GetSortedCutCopySet**
  Returns a sorted list of nodes, which are marked for cut or copy clipboard operation.

- **GetSortedSelection**
  Returns a sorted list of all currently selected nodes.

- **GetTextInfo**
  Helper method for node editors, hints etc.

- **GetTreeFromDataObject**
  OLE drag'n drop and clipboard support method.

- **GetTreeRect**
  Returns the size of the virtual tree image.

- **GetVisibleParent**
  Returns the first (nearest) parent node, which is visible.

- **HandleHotTrack**
  Not documented.
HandleIncrementalSearch
Not documented.

HandleMouseDblClick
Not documented.

HandleMouseDown
Not documented.

HandleMouseUp
Not documented.

HasAsParent
Determines if the given node has got another node as one of its parents.

HasImage
Not documented.

HasPopupMenu
Determines whether there is a pop up menu assigned to the tree.

InitChildren
Not documented.

InitNode
Not documented.

InsertNode
Inserts a new node and returns it to the caller.

InternalAddFromStream
Not documented.

InternalAddToSelection
Not documented.

InternalCacheNode
Not documented.

InternalClearSelection
Not documented.

InternalConnectNode
Not documented.

InternalData
Returns the address of the internal data for a tree class.

InternalDisconnectNode
Not documented.
- **InternalRemoveFromSelection**
  Not documented.
- **InvalidateCache**
  Empties the internal node cache and marks it as invalid.
- **InvalidateChildren**
  Invalidates all children of the given node.
- **InvalidateColumn**
  Invalidates the client area part of a column.
- **InvalidateNode**
  Invalidates the given node.
- **InvalidateToBottom**
  Invalidates the client area starting with the top position of the given node.
- **InvertSelection**
  Inverts the current selection.
- **IsEditing**
  Tells the caller whether the tree is currently in edit mode.
- **IsMouseSelecting**
  Tell the caller whether the tree is currently in draw selection mode.
- **IterateSubtree**
  Iterator method to go through all nodes of a given sub tree.
- **Loaded**
  Not documented.
- **LoadFromFile**
  Loads previously streamed out tree data back in again.
- **LoadFromStream**
  Loads previously streamed out tree data back in again.
- **MainColumnChanged**
  Not documented.
- **MarkCutCopyNodes**
  Not documented.
- **MeasureItemHeight**
  Not documented.
- **MouseMove**
  Not documented.
MoveTo
Moves Source and all its child nodes to Target.

Notification
Not documented.

OriginalWMNCPaint
Not documented.

Paint
TControl's Paint method used here to display the tree.

PaintCheckImage
Not documented.

PaintImage
Not documented.

PaintNodeButton
Not documented.

PaintSelectionRectangle
Not documented.

PaintTree
Main paint routine for the tree image.

PaintTreeLines
Not documented.

PanningWindowProc
Not documented.

PasteFromClipboard
Inserts the content of the clipboard into the tree.

PrepareDragImage
Not documented.

Print
Not documented.

ProcessDrop
Helper method to ease OLE drag'n drop operations.

ProcessOLEData
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk
Not documented.

ReadNode
- Not documented.
- **RedirectFontChangeEvent**
  - Not documented.
- **ReinitChildren**
  - Forces all child nodes of Node to be reinitialized.
- **ReinitNode**
  - Forces a reinitialization of the given node.
- **RemoveFromSelection**
  - Removes the given node from the current selection.
- **RenderOLEData**
  - Renders pending OLE data.
- **RepaintNode**
  - Causes the treeview to repaint the given node.
- **ResetNode**
  - Resets the given node to uninitialized.
- **ResetRangeAnchor**
  - Not documented.
- **RestoreFontChangeEvent**
  - Not documented.
- **SaveToFile**
  - Saves the entire content of the tree into a file or stream.
- **SaveToStream**
  - Saves the entire content of the tree into a file or stream.
- **ScrollIntoView**
  - Scrolls the tree so that the given node comes in the client area.
- **SelectAll**
  - Selects all nodes in the tree.
- **SelectNodes**
  - Selects a range of nodes.
- **SetBiDiMode**
  - Not documented.
- **SetFocusedNodeAndColumn**
  - Not documented.
- **SkipNode**
  - Not documented.
Sort
Sorts the given node.

SortTree
Sorts the entire tree view.

StartWheelPanning
Not documented.

StopWheelPanning
Not documented.

StructureChange
Not documented.

SuggestDropEffect
Not documented.

ToggleNode
Changes a node's expand state to the opposite state.

ToggleSelection
Toggles the selection state of a range of nodes.

UnselectNodes
Deselects a range of nodes.

UpdateAction
Not documented.

UpdateDesigner
Not documented.

UpdateEditBounds
Not documented.

UpdateHeaderRect
Not documented.

UpdateHorizontalScrollBar
Applies changes to the horizontal and vertical scrollbars.

UpdateScrollBars
Applies changes to the horizontal and vertical scrollbars.

UpdateVerticalScrollBar
Applies changes to the horizontal and vertical scrollbars.

UpdateWindowAndDragImage
Not documented.

UseRightToLeftReading
Helper method for right-to-left layout.

- **ValidateCache**
  Initiates the validation of the internal node cache.

- **ValidateChildren**
  Validates all children of a given node.

- **ValidateNode**
  Validates a given node.

- **ValidateNodeDataSize**
  Helper method for node data size initialization.

- **WndProc**
  Redirected window procedure to do some special processing.

- **WriteChunks**
  Writes the core chunks for the given node to the given stream.

- **WriteNode**
  Writes the cover (envelop) chunk for the given node to the given stream.
TBaseVirtualTree Class Properties

Properties

- **Alignment**: Determines the horizontal alignment of text if no columns are defined.
- **AnimationDuration**: Determines the maximum duration the tree can use to play an animation.
- **AutoExpandDelay**: Time delay after which a node gets expanded if it is the current drop target.
- **AutoScrollDelay**: Time which determines when auto scrolling should start.
- **AutoScrollInterval**: Time interval between scroll events when doing auto scroll.
- **Background**: Holds a background image for the tree.
- **BackgroundOffsetX**: Horizontal offset of the background image.
- **BackgroundOffsetY**: Vertical offset of the background image.
- **BorderStyle**: Same as TForm.BorderStyle.
- **ButtonFillMode**: Determines how to fill the background of the node buttons.
- **ButtonStyle**: Determines the look of node buttons.
- **ChangeDelay**: Time which determines when the **OnChange** event should be triggered after the actual change event.
- **CheckImageKind**
  Determines which images should be used for checkboxes and radio buttons.

- **CheckImages**
  Not documented.

- **CheckState**
  Read or set the check state of a node.

- **CheckType**
  Read or set the check type of a node.

- **ChildCount**
  Read or set the number of child nodes of a node.

- **ChildrenInitialized**
  Read whether a node's child count has been initialized already.

- **ClipboardFormats**
  Special class to keep a list of clipboard format descriptions.

- **Colors**
  A collection of colors used in the tree.

- **CustomCheckImages**
  Assign your own image list to get the check images you like most.

- **DefaultNodeHeight**
  Read or set the height new nodes get as initial value.

- **DefaultPasteMode**
  Read or set the value, which determines where to add pasted nodes to.

- **DragHeight**
  Read or set the vertical limit of the internal drag image.

- **DragImage**
  Holds the instance of the internal drag image.

- **DragImageKind**
  Read or set what should be shown in the drag image.

- **DragManager**
  Holds the reference to the internal drag manager.

- **DragOperations**
  Read or set which drag operations may be allowed in the tree.

- **DragSelection**
Keeps a temporary list of nodes during drag'n drop.

- **DragType**
  Read or set which subsystem should be used for dragging.

- **DragWidth**
  Read or set the horizontal limit of the internal drag image.

- **DrawSelectionMode**
  Read or set how multiselection with the mouse is to be visualized.

- **DropTargetNode**
  Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

- **EditColumn**
  Not documented.

- **EditDelay**
  Read or set the maximum time between two single clicks on the same node, which should start node editing.

- **EditLink**
  Keeps a reference to the internal edit link during a node edit operation.

- **Expanded**
  Read or set the expanded state of a particular node.

- **FocusedColumn**
  Read or set the currently focused column.

- **FocusedNode**
  Read or set the currently focused node.

- **Font**
  Same as TWinControl.Font.

- **FullyVisible**
  Read or set whether a node is fully visible or not.

- **HasChildren**
  Read or set whether a node has got children.

- **Header**
  Provides access to the header instance.

- **HeaderRect**
  Returns the non-client-area rectangle used for the header.

- **HintAnimation**
Read or set the current hint animation type.

- **HintMode**
  Read or set what type of hint you want for the tree view.

- **HotCursor**
  Read or set which cursor should be used for hot nodes.

- **HotNode**
  Read, which node is currently the hot node.

- **Images**
  Read or set the tree's normal image list.

- **IncrementalSearch**
  Read or set the current incremental search mode.

- **IncrementalSearchDirection**
  Read or set the direction to be used for incremental search.

- **IncrementalSearchStart**
  Read or set where to start incremental search.

- **IncrementalSearchTimeout**
  Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

- **Indent**
  Read or set the indentation amount for node levels.

- **IsDisabled**
  Read or set the enabled state of the given node.

- **IsVisible**
  Read or set the visibility state of the given node.

- **LastClickPos**
  Used for retained drag start and wheel mouse scrolling.

- **LastDropMode**
  Read how the last drop operation finished.

- **LineMode**
  Read or set the mode of the tree lines.

- **LineStyle**
  Read or set the mode of the tree lines.

- **Margin**
  Read or set the tree's node margin.

- **MultiLine**
Read or toggle the multiline feature for a given node.

- **NodeAlignment**
  Read or set the node alignment value.

- **NodeDataSize**
  Read or set the extra data size for each node.

- **NodeHeight**
  Read or set a node's height.

- **NodeParent**
  Read or set a node's parent node.

- **OffsetX**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetXY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **RootNode**
  Reference to the internal root node which is the anchor of the entire tree node hierarchy.

- **RootNodeCount**
  Read or set the number of nodes on the top level.

- **ScrollBarOptions**
  Reference to the scroll bar options class.

- **SearchBuffer**
  Current input string for incremental search.

- **Selected**
  Property to modify or determine the selection state of a node.

- **SelectedCount**
  Contains the number of selected nodes.

- **SelectionBlendFactor**
  Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

- **SelectionCurveRadius**
  Read or set the current corner radius for node selection rectangles.

- **StateImages**
  Reference to the images list which is used for the state images.
- **TextMargin**
  Read or set the distance of the node caption to its borders.

- **TopNode**
  The top node is the node which is currently at the top border of the client area.

- **TotalCount**
  Returns the number of nodes in the tree.

- **TotalInternalDataSize**
  Keeps the currently accumulated data size for one node.

- **TreeOptions**
  Reference to the tree's options.

- **TreeStates**
  Property which keeps a set of flags which indicate current operation and states of the tree.

- **UpdateCount**
  Not documented.

- **VerticalAlignment**
  Used to set a node's vertical button alignment with regard to the entire node rectangle.

- **VisibleCount**
  Number of currently visible nodes.

- **VisiblePath**
  Property to set or determine a node parent's expand states.

- **WantTabs**
  Read or set whether the tree wants to process tabs on its own.
See Also
TBaseVirtualTree.CustomCheckImages Property

See Also
TCheckImageKind
See Also
TBaseVirtualTree.DefaultPasteMode Property

See Also
TVTNodeAttachMode
See Also
TBaseVirtualTree.DragManager Property

See Also
TVTDragManager
See Also
TBaseVirtualTree.EditDelay Property

See Also
Editors and editing
See Also
TBaseVirtualTree.FocusedColumn Property

See Also
FocusedNode, TVTSelectionOptions
See Also
TBaseVirtualTree.FocusedNode Property

See Also
FocusedColumn, TVTSelectionOptions
See Also

TBaseVirtualTree.FullyVisible Property

See Also

IsVisible, VisiblePath, vsVisible, TVirtualNodeStates
See Also
TBaseVirtualTree.HasChildren Property

See Also
vsHasChildren, TVirtualNodeStates
See Also
TBaseVirtualTree.Header Property

See Also
TVTHeader
See Also
TBaseVirtualTree.HotCursor Property

See Also
HotNode, TVTPaintOptions
See Also
TBaseVirtualTree.HotNode Property

See Also
HotCursor, toHotTrack, TVTPaintOptions
See Also
TBaseVirtualTree.Images Property

See Also
StateImages, CheckImages
See Also
TBaseVirtualTree.IncrementalSearch Property

See Also
IncrementalSearchDirection, IncrementalSearchStart, IncrementalSearchTimeout
See Also
TBaseVirtualTree.IncrementalSearchDirection Property

See Also
IncrementalSearch, IncrementalSearchStart, IncrementalSearchTime123out
See Also
TBaseVirtualTree.IncrementalSearchStart Property

See Also
IncrementalSearch, IncrementalSearchDirection, IncrementalSearchTimeout
See Also
TBaseVirtualTree.IncrementalSearchTimeout Property

See Also
IncrementalSearch, IncrementalSearchDirection, IncrementalSearchStart
See Also
TBaseVirtualTree.Margin Property

See Also
TVirtualStringTree.TextMargin
See Also
TBaseVirtualTree.NodeAlignment Property

See Also
TVirtualNode
See Also
TBaseVirtualTree.NodeDataSize Property

See Also
Data handling
See Also
TBaseVirtualTree.NodeParent Property

See Also
MoveTo, CopyTo
See Also

TBaseVirtualTree.OnAdvancedHeaderDraw Event

See Also

OnHeaderDrawQueryElements, OnHeaderDraw
See Also
TBaseVirtualTree.OnAfterCellPaint Event

See Also
Paint cycles and stages
See Also
TBaseVirtualTree.OnAfterItemErase Event

See Also
Paint cycles and stages
See Also
TBaseVirtualTree.OnAfterItemPaint Event

See Also
Paint cycles and stages
See Also

TBaseVirtualTree.OnAfterPaint Event

See Also

Paint cycles and stages
See Also
TBaseVirtualTree.OnBeforeCellPaint Event

See Also
Paint cycles and stages
See Also
TBaseVirtualTree.OnBeforeItemErase Event

See Also
Paint cycles and stages
See Also
TBaseVirtualTree.OnBeforeItemPaint Event

See Also
Paint cycles and stages
See Also
TBaseVirtualTree.OnBeforePaint Event

See Also
Paint cycles and stages
See Also
TBaseVirtualTree.OnColumnClick Event

See Also
OnHeaderClick
See Also

TBaseVirtualTree.OnColumnDbIClick Event

See Also

OnColumnClick, OnHeaderDbIClick
See Also
TBaseVirtualTree.OnCompareNodes Event

See Also
SortTree, Sort
See Also
TBaseVirtualTree.OnCreateEditor Event

See Also
Editors and editing
See Also
TBaseVirtualTree.OnDragOver Event

See Also
OnDragDrop
See Also
TBaseVirtualTree.OnEditCancelled Event

See Also
Editors and editing
See Also
TBaseVirtualTree.OnEdited Event

See Also
Editors and editing
See Also

TBaseVirtualTree.OnEditing Event

See Also

Editors and editing
See Also
TBaseVirtualTree.OnGetLineStyle Event

See Also
PrepareBitmaps
See Also
TBaseVirtualTree.OnGetNodeDataSize Event

See Also
NodeDataSize, Data handling
See Also
TBaseVirtualTree.OnHeaderClick Event

See Also
SortColumn, SortDirection
See Also
TBaseVirtualTree.OnHeaderDblClick Event

See Also
OnHeaderClick
See Also
TBaseVirtualTree.OnHeaderDrawQueryElements Event

See Also
OnAdvancedHeaderDraw
See Also
TBaseVirtualTree.OnInitChildren Event

See Also
The virtual paradigm
See Also
TBaseVirtualTree.OnInitNode Event

See Also
The virtual paradigm
See Also
TBaseVirtualTree.OnLoadNode Event

See Also
OnSaveNode, LoadFromStream, SaveToStream, AddFromStream, VTTreeStreamVersion, TVTHeader.LoadFromStream, TVTHeader.SaveToStream
See Also
TBaseVirtualTree.OnMeasureItem Event

See Also
Invalidatenode, vsHeightMeasured
See Also
TBaseVirtualTree.OnResetNode Event

See Also
ResetNode
See Also

TBaseVirtualTree.OnSaveNode Event

See Also

OnLoadNode, LoadFromStream, SaveToStream, AddFromStream, VTTreeStreamVersion, TVTHeader.LoadFromStream, TVTHeader.SaveToStream
See Also
TBaseVirtualTree.OnScroll Event

See Also
OffsetXY
See Also

TBaseVirtualTree.SearchBuffer Property

See Also

IncrementalSearch
See Also
TBaseVirtualTree.SelectionBlendFactor Property

See Also
DrawSelectionMode, TVTPaintOptions
See Also
TBaseVirtualTree.SelectionCurveRadius Property

See Also
SelectionBlendFactor, DrawSelectionMode, TVTPaintOptions
See Also
TBaseVirtualTree.StateImages Property

See Also
CheckImages, Images
See Also
TBaseVirtualTree.TextMargin Property

See Also
Margin
See Also
TBaseVirtualTree.TotalInternalDataSize Property

See Also
Data handling
See Also
TBaseVirtualTree.TreeStates Property

See Also
OnStateChange
See Also
TBaseVirtualTree.VisiblePath Property

See Also
Visible
See Also
TBaseVirtualTree.AddChild Method

See Also
InsertNode, OnInitNode, OnInitChildren
See Also
TBaseVirtualTree.AddFromStream Method

See Also
SaveToStream
See Also
TBaseVirtualTree.AllocateInternalDataArea Method

See Also
Data handling, TotalInternalDataSize
See Also
TBaseVirtualTree.Change Method

See Also
BeginSynch, EndSynch, BeginUpdate, EndUpdate, ChangeDelay
See Also
TBaseVirtualTree.ChangeScale Method

See Also
TVTHeader.ChangeScale, DefaultNodeHeight
See Also
TBaseVirtualTree.ColumnIsEmpty Method

See Also
toAutoSpanColumns
See Also
TBaseVirtualTree.DetermineHiddenChildrenFlag Method

See Also
vsVisible, toAutoHideButtons
See Also
TBaseVirtualTree.DoEdit Method

See Also
tsEditing, OnCreateEditor, IVTEditLink
See Also
TBaseVirtualTree.DoEndEdit Method

See Also
DoEdit, OnNewText, EditNode
See Also
TBaseVirtualTree.EditNode Method

See Also
DoEdit
See Also
TBaseVirtualTree.EndEditNode Method

See Also
Editors and editing, EditNode, DoEdit
See Also
TBaseVirtualTree.EndSynch Method

See Also
BeginSynch, BeginUpdate, EndUpdate
See Also
TBaseVirtualTree.FullCollapse Method

See Also
   FullExpand
See Also
TBaseVirtualTree.InternalData Method

See Also
Data handling
See Also
TBaseVirtualTree.LoadFromFile Method

See Also
AddFromStream
See Also
TBaseVirtualTree.PaintTree Method

See Also
Tree image and tree window
See Also
TBaseVirtualTree.SaveToFile Method

See Also
LoadFromStream, AddFromStream
See Also

TBaseVirtualTree.ValidateCache Method

See Also

InvalidateCache
See Also
TBaseVirtualTree.WriteChunks Method

See Also
WriteNode, SaveToStream
See Also
TBaseVirtualTree.WriteNode Method

See Also
WriteChunks, WriteToStream
TBufferedString Class Methods

Methods

- Add
  Not documented.
- AddNewLine
  Not documented.
- Destroy
  Not documented.
### TBufferedString Class Properties

**TBufferedString Class | Legend**

#### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AsString</code></td>
<td>Not documented.</td>
</tr>
</tbody>
</table>
Legend

TBufferedString Class

Legend

- public
- Property
- read only
- Method
- virtual
TClipboardFormatList Class Methods

Methods

- **Add**
  Adds the given data to the internal list.

- **Clear**
  Not documented.

- **Create**
  Not documented.

- **Destroy**
  Not documented.

- **EnumerateFormats**
  Returns a list of format records for the given class.

- **FindFormat**
  Not documented.
TClipboardFormats Class Methods

Methods

- **Add**
  Adds a new format to the internal list.

- **Create**
  Constructor of the class.

- **Insert**
  Adds a new format to the internal list.
<table>
<thead>
<tr>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
</tr>
</tbody>
</table>
Legend
TClipboardFormats Class

Legend

public

Property

read only

Method

virtual
TCriticalSection Class Fields

| TCriticalSection Class | Legend |

**Fields**

- **FSection**
  
  Not documented.
TCriticalSection Class Methods

TCriticalSection Class | Legend

Methods

- Create
  Not documented.
- Destroy
  Not documented.
- Enter
  Not documented.
- Leave
  Not documented.
Legend

TCriticalSection Class

Legend

- protected
- Data Member
- public
- Method
- virtual
TCustomStringTreeOptions Class Methods

Methods

- ![AssignTo] AssignTo
  Used to copy the options class.
- ![Create] Create
  The constructor of the class.

TCustomVirtualTreeOptions Class

- ![AssignTo] AssignTo
  Used to copy this option class to another option collection.
- ![Create] Create
  Constructor of the class.
TCustomStringTreeOptions Class Properties

Properties

- **StringOptions**
  The new options introduced by the class.

TCustomVirtualTreeOptions Class

- **AnimationOptions**
  Options related to animations.
- **AutoOptions**
  Options related to automatic actions.
- **MiscOptions**
  Options not related to any other category.
- **Owner**
  Owner tree to which the property class belongs.
- **PaintOptions**
  Options related to painting.
- **SelectionOptions**
  Options related to the way nodes can be selected.
Legend

TCustomStringTreeOptions Class

Legend

- protected
- Property
- public
- read only
- Method
- virtual
TCustomVirtualDrawTree Class Events

**Events**

- **OnDrawHint**
  Triggered when a node hint or tooltip must be drawn.
- **OnDrawNode**
  Triggered when a node must be drawn.
- **OnGetHintSize**
  Triggered when a node hint or tooltip is about to show.
- **OnGetNodeWidth**
  Triggered when a node is about to be drawn.

TBaseVirtualTree Class

- **OnAdvancedHeaderDraw**
  Header paint support event.
- **OnAfterCellPaint**
  Paint support event.
- **OnAfterItemErase**
  Paint support event.
- **OnAfterItemPaint**
  Paint support event.
- **OnAfterPaint**
  Paint support event.
- **OnBeforeCellPaint**
  Paint support event.
- **OnBeforeItemErase**
  Paint support event.
- **OnBeforeItemPaint**
  Paint support event.
OnBeforePaint
Paint support event.

OnChange
Navigation support event.

OnChecked
Check support event.

OnChecking
Check support event.

OnCollapsed
Miscellaneous event.

OnCollapsing
Miscellaneous event.

OnColumnClick
Header and column support event.

OnColumnDblClick
Header and column support event.

OnColumnResize
Header and column support routine.

OnCompareNodes
Sort and search support event.

OnCreateDataObject
Drag'n drop support event.

OnCreateDragManager
Drag'n drop support event.

OnCreateEditor
Editing support event.

OnDragAllowed
Drag'n drop support event.

OnDragDrop
Drag'n drop support event.

OnDragOver
Drag'n drop support event.

OnEditCancelled
Editing support event.

OnEdited
Editing support event.
- **OnEditing**
  Editing support event.
- **OnExpanded**
  Miscellaneous event.
- **OnExpanding**
  Miscellaneous event.
- **OnFocusChanged**
  Navigation support event.
- **OnFocusChanging**
  Navigation support event.
- **OnFreeNode**
  Data management node.
- **OnGetCellIsEmpty**
  Triggered when the tree control needs to know whether a given column is empty.
- **OnGetCursor**
  Miscellaneous event.
- **OnGetHeaderCursor**
  Header and column support event.
- **OnGetHelpContext**
  Miscellaneous event.
- **OnGetImageIndex**
  Display management event.
- **OnGetImageIndexEx**
  Not documented.
- **OnGetLineStyle**
  Display management event.
- **OnGetNodeDataSize**
  Data management event.
- **OnGetPopupMenu**
  Miscellaneous event.
- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.
- **OnHeaderClick**
Header & column support event.
- OnHeaderDb1Click
Header & column support event.
- OnHeaderDragged
Header & column support event.
- OnHeaderDraggedOut
Header & column support event.
- OnHeaderDragging
Header & column support event.
- OnHeaderDraw
Header & column support event.
- OnHeaderDrawQueryElements
Header & column support event.
- OnHeaderMouseDown
Header & column support event.
- OnHeaderMouseMove
Header & column support event.
- OnHeaderMouseUp
Header & column support event.
- OnHotChange
Navigation support event.
- OnIncrementalSearch
Miscellaneous event.
- OnInitChildren
Node management event.
- OnInitNode
Node management event.
- OnKeyAction
Miscellaneous event.
- OnLoadNode
Streaming support event.
- OnMeasureItem
Miscellaneous event.
- OnNodeCopied
Miscellaneous event.
OnNodeCopying
  Miscellaneous event.
OnNodeMoved
  Miscellaneous event.
OnNodeMoving
  Miscellaneous event.
OnPaintBackground
  Paint support event.
OnRenderOLEData
  Drag'n drop and clipboard support event.
OnResetNode
  Node management event.
OnSaveNode
  Streaming support event.
OnScroll
  Miscellaneous event.
OnShowScrollbar
  Not documented.
OnStateChange
  Miscellaneous event.
OnStructureChange
  Miscellaneous event.
OnUpdating
  Miscellaneous event.
TCustomVirtualDrawTree Class Methods

**Methods**

- **DoDrawHint**
  Overridable method which triggers `OnDrawHint`.
- **DoGetHintSize**
  Overridable method which triggers `OnGetHintSize`.
- **DoGetNodeWidth**
  Overridable method which triggers `OnGetNodeWidth`.
- **DoPaintNode**
  Overridable method which triggers `OnPaintNode`.

**TBaseVirtualTree Class**

- **AbsoluteIndex**
  Reads the overall index of a node.
- **AddChild**
  Creates and adds a new child node to given node.
- **AddFromStream**
  Adds the content from the given stream to the given node.
- **AddToSelection**
  Adds one or more nodes to the current selection.
- **AdjustPaintCellRect**
  Used in descendants to modify the clip rectangle of the current column while painting a certain node.
- **AdjustPanningCursor**
  Loads the proper cursor which indicates into which direction scrolling is done.
- **AdviseChangeEvent**
  Used to register a delayed change event.
AllocateInternalDataArea
Registration method to allocate tree internal data per node.

Animate
Support method for animated actions in the tree view.

Assign
Used to copy properties from another Virtual Treeview.

BeginDrag
Starts an OLE drag'n drop operation.

BeginSynch
Enters the tree into a special synchronized mode.

BeginUpdate
Locks the tree view to perform several update operations.

CalculateSelectionRect
Support method for draw selection.

CanAutoScroll
Determines whether the tree can currently auto scroll its window.

CancelCutOrCopy
Cancels any pending cut or copy clipboard operation.

CancelEditNode
Cancel the current edit operation, if there is any.

CanEdit
Determines whether a node can be edited or not.

CanFocus
Support method to determine whether the tree window can receive the input focus.

CanShowDragImage
Determines whether a drag image should be shown.

Change
Central method called when a node's selection state changes.

ChangeScale
Helper method called by the VCL when control resizing is due.

CheckParentCheckState
Helper method for recursive check state changes.

Clear
Clears the tree and removes all nodes.
ClearChecked
Not documented.

ClearSelection
Removes all nodes from the current selection.

ClearTempCache
Helper method to clear the internal temporary node cache.

ColumnIsEmpty
Used to determine if a cell is considered as being empty.

CopyTo
Copies Source and all its child nodes to Target.

CopyToClipBoard
Copies all currently selected nodes to the clipboard.

CountLevelDifference
Determines the level difference of two nodes.

CountVisibleChildren
Determines the number of visible child nodes of the given node.

Create
Constructor of the control

CreateParams
Prepares the creation of the controls window handle.

CreateWnd
Initializes data, which depends on the window handle.

CutToClipBoard
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties
Helper method to customize loading and saving persistent tree data.

DeleteChildren
Removes all child nodes from the given node.

DeleteNode
Removes the given node from the tree.

DeleteSelectedNodes
Removes all currently selected nodes from the tree.

Destroy
Destructor of the control.
DetermineHiddenChildrenFlag
Determine whether all children of a given node are hidden.

DetermineHiddenChildrenFlagAllNodes
Determine whether all children of all nodes are hidden.

DetermineHitPositionLTR
Determine the hit position within a node with left-to-right and right-to-left orientation.

DetermineHitPositionRTL
Determine the hit position within a node with left-to-right and right-to-left orientation.

DetermineNextCheckState
Not documented.

DetermineScrollDirections
Not documented.

DoAdvancedHeaderDraw
Not documented.

DoAfterCellPaint
Not documented.

DoAfterItemErase
Not documented.

DoAfterItemPaint
Not documented.

DoBeforeCellPaint
Not documented.

DoBeforeDrag
Not documented.

DoBeforeItemErase
Not documented.

DoBeforeItemPaint
Not documented.

DoBeforePaint

DoAutoScroll
Enables or disables the auto scroll timer.
Not documented.

- **DoCancelEdit**
  Called when the tree should stop editing without accepting changed values.

- **DoCanEdit**
  Not documented.

- **DoChange**
  Not documented.

- **DoCheckClick**
  Not documented.

- **DoChecked**
  Not documented.

- **DoChecking**
  Not documented.

- **DoCollapsed**
  Not documented.

- **DoCollapsing**
  Not documented.

- **DoColumnClick**
  Not documented.

- **DoColumnDbIClick**
  Not documented.

- **DoColumnResize**
  Not documented.

- **DoCompare**
  Not documented.

- **DoCreateDataObject**
  Not documented.

- **DoCreateDragManager**
  Not documented.

- **DoCreateEditor**
  Not documented.

- **DoDragDrop**
  Not documented.

- **DoDragExpand**
Not documented.

DoDragging
Internal method which handles drag' drop.

DoDragOver
Not documented.

DoEdit
Initiates editing of the currently set focused column and edit node.

DoEndDrag
Not documented.

DoEndEdit
Stops the current edit operation and takes over the new content.

DoExpanded
Not documented.

DoExpanding
Not documented.

DoFocusChange
Not documented.

DoFocusChanging
Not documented.

DoFocusNode
Internal method to set the focused node.

DoFreeNode
Not documented.

DoGetAnimationType
Determines the type of animation to be used.

DoGetCursor
Not documented.

DoGetHeaderCursor
Not documented.

DoGetImageIndex
Not documented.

DoGetLineStyle
Not documented.

DoGetNodeHint
Not documented.
DoGetNodeTooltip
Not documented.

DoGetNodeWidth
Overridable method which always returns 0.

DoGetPopupMenu
Overridable method which triggers the OnGetPopup event.

DoGetUserClipboardFormats
Not documented.

DoHeaderClick
Not documented.

DoHeaderDoubleClick
Not documented.

DoHeaderDragged
Not documented.

DoHeaderDraggedOut
Not documented.

DoHeaderDragging
Not documented.

DoHeaderDraw
Not documented.

DoHeaderDrawQueryElements
Not documented.

DoHeaderMouseDown
Not documented.

DoHeaderMouseMove
Not documented.

DoHeaderMouseUp
Not documented.

DoHotChange
Not documented.

DoIncrementalSearch
Not documented.

DoInitChildren
Not documented.

DoInitNode
Not documented.

- **DoKeyAction**
  Not documented.

- **DoLoadUserData**
  Not documented.

- **DoMeasureItem**
  Not documented.

- **DoNodeCopied**
  Not documented.

- **DoNodeCopying**
  Not documented.

- **DoNodeMoved**
  Not documented.

- **DoNodeMoving**
  Not documented.

- **DoPaintBackground**
  Not documented.

- **DoPaintDropMark**
  Overridable method which draws the small line on top of a nodes image depending on the current drop state.

- **DoPaintNode**
  Overridable method which does nothing.

- **DoPopupMenu**
  Overridable method which shows the popup menu for the given node.

- **DoRenderOLEData**
  Not documented.

- **DoReset**
  Not documented.

- **DoSaveUserData**
  Not documented.

- **DoScroll**
  Overridable method which triggers the **OnScroll** event.

- **DoSetOffsetXY**
  Internal core routine to set the tree's scroll position.

- **DoShowScrollbar**
DoStartDrag
Not documented.

DoStateChange
Not documented.

DoStructureChange
Not documented.

DoTimerScroll
Callback method which is triggered whenever the scroll timer fires.

DoUpdating
Not documented.

DoValidateCache
Not documented.

DragCanceled
Called by the VCL when a drag'n drop operation was canceled by the user.

DragDrop
Helper method, which is used when a drag operation is finished.

DragEnter
Not documented.

DragFinished
Called when a drag operation is finished (accepted or cancelled).

Dragging
Returns true if a drag'n drop operation is in progress.

DragLeave
Not documented.

DragOver
Not documented.

DrawDottedHLine
Not documented.

DrawDottedVLine
Not documented.

EditNode
Starts editing the given node if allowed to.

EndEditNode
Stops node editing if it was started before.

EndSynch
Counterpart to BeginSynch.

EndUpdate
Resets the update lock set by BeginUpdate.

ExecuteAction
Not documented.

FindNodeInSelection
Helper method to find the given node in the current selection.

FinishChunkHeader
Not documented.

FinishCutOrCopy
Stops any pending cut or copy clipboard operation.

FlushClipboard
Renders all pending clipboard data.

FontChanged
Not documented.

FullCollapse
Collapses all nodes in the tree.

FullExpand
Expands all nodes in the tree.

GetBorderDimensions
Not documented.

GetCheckImage
Not documented.

GetCheckImageListFor
Not documented.

GetColumnClass
Returns the class to be used to manage columns in the tree.

GetControlsAlignment
Not documented.

GetDisplayRect
Returns the visible region used by the given node in client coordinates.

GetFirst
Group of node navigation functions.

- **GetFirstChecked**
  Not documented.

- **GetFirstChild**
  Group of node navigation functions.

- **GetFirstCutCopy**
  Group of node navigation functions.

- **GetFirstInitialized**
  Group of node navigation functions.

- **GetFirstNoInit**
  Group of node navigation functions.

- **GetFirstSelected**
  Group of node navigation functions.

- **GetFirstVisible**
  Group of node navigation functions.

- **GetFirstVisibleChild**
  Group of node navigation functions.

- **GetFirstVisibleChildNoInit**
  Group of node navigation functions.

- **GetFirstVisibleNoInit**
  Group of node navigation functions.

- **GetHeaderClass**
  Returns the header class to be used by the tree.

- **GetHintWindowClass**
  Not documented.

- **GetHitTestInfoAt**
  Returns information about the node at the given position.

- **GetImageIndex**
  Not documented.

- **GetLast**
  Group of node navigation functions.

- **GetLastChild**
  Group of node navigation functions.

- **GetLastChildNoInit**
  Group of node navigation functions.
GetLastInitialized
Group of node navigation functions.

GetLastNoInit
Group of node navigation functions.

GetLastVisible
Group of node navigation functions.

GetLastVisibleChild
Group of node navigation functions.

GetLastVisibleChildNoInit
Group of node navigation functions.

GetLastVisibleNoInit
Group of node navigation functions.

GetMaxColumnWidth
Returns the width of the largest node in the given column.

GetMaxRightExtend
Determines the maximum width of the currently visible part of the tree.

GetNativeClipboardFormats
Used to let descendants and the application add their own supported clipboard formats.

GetNext
Group of node navigation functions.

GetNextChecked
Not documented.

GetNextCutCopy
Group of node navigation functions.

GetNextInitialized
Group of node navigation functions.

GetNextNoInit
Group of node navigation functions.

GetNextSelected
Group of node navigation functions.

GetNextSibling
Group of node navigation functions.

GetNextVisible
Group of node navigation functions.
- GetNextVisibleNoInit
  Group of node navigation functions.
- GetNextVisibleSibling
  Group of node navigation functions.
- GetNextVisibleSiblingNoInit
  Group of node navigation functions.
- GetNodeAt
  Not documented.
- GetNodeData
  Returns the address of the user data area of the given node.
- GetNodeLevel
  Returns the indentation level of the given node.
- GetOptionsClass
  Customization helper to determine which options class the tree should use.
- GetPrevious
  Group of node navigation functions.
- GetPreviousInitialized
  Group of node navigation functions.
- GetPreviousNoInit
  Group of node navigation functions.
- GetPreviousSibling
  Group of node navigation functions.
- GetPreviousVisible
  Group of node navigation functions.
- GetPreviousVisibleNoInit
  Group of node navigation functions.
- GetPreviousVisibleSibling
  Group of node navigation functions.
- GetPreviousVisibleSiblingNoInit
  Group of node navigation functions.
- GetSortedCutCopySet
  Returns a sorted list of nodes, which are marked for cut or copy clipboard operation.
- GetSortedSelection
Returns a sorted list of all currently selected nodes.

- **GetTextInfo**
  Helper method for node editors, hints etc.

- **GetTreeFromDataObject**
  OLE drag'n drop and clipboard support method.

- **GetTreeRect**
  Returns the size of the virtual tree image.

- **GetVisibleParent**
  Returns the first (nearest) parent node, which is visible.

- **HandleHotTrack**
  Not documented.

- **HandleIncrementalSearch**
  Not documented.

- **HandleMouseDb1Click**
  Not documented.

- **HandleMouseDown**
  Not documented.

- **HandleMouseUp**
  Not documented.

- **HasAsParent**
  Determines if the given node has got another node as one of its parents.

- **HasImage**
  Not documented.

- **HasPopupMenu**
  Determines whether there is a pop up menu assigned to the tree.

- **InitChildren**
  Not documented.

- **InitNode**
  Not documented.

- **InsertNode**
  Inserts a new node and returns it to the caller.

- **InternalAddFromStream**
  Not documented.

- **InternalAddToSelection**
Not documented.

- InternalCacheNode
  Not documented.

- InternalClearSelection
  Not documented.

- InternalConnectNode
  Not documented.

- InternalData
  Returns the address of the internal data for a tree class.

- InternalDisconnectNode
  Not documented.

- InternalRemoveFromSelection
  Not documented.

- InvalidateCache
  Empties the internal node cache and marks it as invalid.

- InvalidateChildren
  Invalidates all children of the given node.

- InvalidateColumn
  Invalidates the client area part of a column.

- InvalidateNode
  Invalidates the given node.

- InvalidateToBottom
  Invalidates the client area starting with the top position of the given node.

- InvertSelection
  Inverts the current selection.

- IsEditing
  Tells the caller whether the tree is currently in edit mode.

- IsMouseSelecting
  Tell the caller whether the tree is currently in draw selection mode.

- IterateSubtree
  Iterator method to go through all nodes of a given sub tree.

- Loaded
  Not documented.

- LoadFromFile
Loads previously streamed out tree data back in again.

`LoadFromStream`

Loads previously streamed out tree data back in again.

`MainColumnChanged`
Not documented.

`MarkCutCopyNodes`
Not documented.

`MeasureItemHeight`
Not documented.

`MouseMove`
Not documented.

`MoveTo`
Moves `Source` and all its child nodes to `Target`.

`Notification`
Not documented.

`OriginalWMNCPaint`
Not documented.

`Paint`
TControl's Paint method used here to display the tree.

`PaintCheckImage`
Not documented.

`PaintImage`
Not documented.

`PaintNodeButton`
Not documented.

`PaintSelectionRectangle`
Not documented.

`PaintTree`
Main paint routine for the tree image.

`PaintTreeLines`
Not documented.

`PanningWindowProc`
Not documented.

`PasteFromClipboard`
Inserts the content of the clipboard into the tree.
PrepareDragImage
Not documented.

Print
Not documented.

ProcessDrop
Helper method to ease OLE drag'n drop operations.

ProcessOLEData
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk
Not documented.

ReadNode
Not documented.

RedirectFontChangeEvent
Not documented.

ReinitChildren
Forces all child nodes of Node to be reinitialized.

ReinitNode
Forces a reinitialization of the given node.

RemoveFromSelection
Removes the given node from the current selection.

RenderOLEData
Renders pending OLE data.

RepaintNode
Causes the treeview to repaint the given node.

ResetNode
Resets the given node to uninitialized.

ResetRangeAnchor
Not documented.

RestoreFontChangeEvent
Not documented.

SaveToFile
Saves the entire content of the tree into a file or stream.

SaveToStream
Saves the entire content of the tree into a file or stream.

ScrollIntoView
Scrolls the tree so that the given node comes in the client area.

- **SelectAll**
  Selects all nodes in the tree.

- **SelectNodes**
  Selects a range of nodes.

- **SetBiDiMode**
  Not documented.

- **SetFocusedNodeAndColumn**
  Not documented.

- **SkipNode**
  Not documented.

- **Sort**
  Sorts the given node.

- **SortTree**
  Sorts the entire tree view.

- **StartWheelPanning**
  Not documented.

- **StopWheelPanning**
  Not documented.

- **StructureChange**
  Not documented.

- **SuggestDropEffect**
  Not documented.

- **ToggleNode**
  Changes a node's expand state to the opposite state.

- **ToggleSelection**
  Toggles the selection state of a range of nodes.

- **UnselectNodes**
  Deselects a range of nodes.

- **UpdateAction**
  Not documented.

- **UpdateDesigner**
  Not documented.

- **UpdateEditBounds**
  Not documented.
- **UpdateHeaderRect**
  Not documented.
- **UpdateHorizontalScrollBar**
  Applies changes to the horizontal and vertical scrollbars.
- **UpdateScrollBars**
  Applies changes to the horizontal and vertical scrollbars.
- **UpdateVerticalScrollBar**
  Applies changes to the horizontal and vertical scrollbars.
- **UpdateWindowAndDragImage**
  Not documented.
- **UseRightToLeftReading**
  Helper method for right-to-left layout.
- **ValidateCache**
  Initiates the validation of the internal node cache.
- **ValidateChildren**
  Validates all children of a given node.
- **ValidateNode**
  Validates a given node.
- **ValidateNodeDataSize**
  Helper method for node data size initialization.
- **WndProc**
  Redirected window procedure to do some special processing.
- **WriteChunks**
  Writes the core chunks for the given node to the given stream.
- **WriteNode**
  Writes the cover (envelop) chunk for the given node to the given stream.
TCustomVirtualDrawTree Class Properties

TBaseVirtualTree Class

- **Alignment**
  Determines the horizontal alignment of text if no columns are defined.

- **AnimationDuration**
  Determines the maximum duration the tree can use to play an animation.

- **AutoExpandDelay**
  Time delay after which a node gets expanded if it is the current drop target.

- **AutoScrollDelay**
  Time which determines when auto scrolling should start.

- **AutoScrollInterval**
  Time interval between scroll events when doing auto scroll.

- **Background**
  Holds a background image for the tree.

- **BackgroundOffsetX**
  Horizontal offset of the background image.

- **BackgroundOffsetY**
  Vertical offset of the background image.

- **BorderStyle**
  Same as TForm.BorderStyle.

- **ButtonFillMode**
  Determines how to fill the background of the node buttons.

- **ButtonStyle**
  Determines the look of node buttons.

- **ChangeDelay**
Time which determines when the **OnChange** event should be triggered after the actual change event.

- **CheckImageKind**
  Determines which images should be used for checkboxes and radio buttons.

- **CheckImages**
  Not documented.

- **CheckState**
  Read or set the check state of a node.

- **CheckType**
  Read or set the check type of a node.

- **ChildCount**
  Read or set the number of child nodes of a node.

- **ChildrenInitialized**
  Read whether a node's child count has been initialized already.

- **ClipboardFormats**
  Special class to keep a list of clipboard format descriptions.

- **Colors**
  A collection of colors used in the tree.

- **CustomCheckImages**
  Assign your own image list to get the check images you like most.

- **DefaultNodeHeight**
  Read or set the height new nodes get as initial value.

- **DefaultPasteMode**
  Read or set the value, which determines where to add pasted nodes to.

- **DragHeight**
  Read or set the vertical limit of the internal drag image.

- **DragImage**
  Holds the instance of the internal drag image.

- **DragImageKind**
  Read or set what should be shown in the drag image.

- **DragManager**
  Holds the reference to the internal drag manager.

- **DragOperations**
Read or set which drag operations may be allowed in the tree.

- **DragSelection**
  Keeps a temporary list of nodes during drag'n drop.

- **DragType**
  Read or set which subsystem should be used for dragging.

- **DragWidth**
  Read or set the horizontal limit of the internal drag image.

- **DrawSelectionMode**
  Read or set how multiselection with the mouse is to be visualized.

- **DropTargetNode**
  Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

- **>EditColumn**
  Not documented.

- **EditDelay**
  Read or set the maximum time between two single clicks on the same node, which should start node editing.

- **EditLink**
  Keeps a reference to the internal edit link during a node edit operation.

- **Expanded**
  Read or set the expanded state of a particular node.

- **FocusedColumn**
  Read or set the currently focused column.

- **FocusedNode**
  Read or set the currently focused node.

- **Font**
  Same as TWinControl.Font.

- **FullyVisible**
  Read or set whether a node is fully visible or not.

- **HasChildren**
  Read or set whether a node has got children.

- **Header**
  Provides access to the header instance.

- **HeaderRect**
Returns the non-client-area rectangle used for the header.

- **HintAnimation**
  Read or set the current hint animation type.

- **HintMode**
  Read or set what type of hint you want for the tree view.

- **HotCursor**
  Read or set which cursor should be used for hot nodes.

- **HotNode**
  Read, which node is currently the hot node.

- **Images**
  Read or set the tree's normal image list.

- **IncrementalSearch**
  Read or set the current incremental search mode.

- **IncrementalSearchDirection**
  Read or set the direction to be used for incremental search.

- **IncrementalSearchStart**
  Read or set where to start incremental search.

- **IncrementalSearchTimeout**
  Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

- **Indent**
  Read or set the indentation amount for node levels.

- **IsDisabled**
  Read or set the enabled state of the given node.

- **IsVisible**
  Read or set the visibility state of the given node.

- **LastClickPos**
  Used for retained drag start and wheel mouse scrolling.

- **LastDropMode**
  Read how the last drop operation finished.

- **LineMode**
  Read or set the mode of the tree lines.

- **LineStyle**
  Read or set the mode of the tree lines.

- **Margin**
Read or set the tree's node margin.

- **MultiLine**
  Read or toggle the multiline feature for a given node.

- **NodeAlignment**
  Read or set the node alignment value.

- **NodeDataSize**
  Read or set the extra data size for each node.

- **NodeHeight**
  Read or set a node's height.

- **NodeParent**
  Read or set a node's parent node.

- **OffsetX**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetXY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **RootNode**
  Reference to the internal root node which is the anchor of the entire tree node hierarchy.

- **RootNodeCount**
  Read or set the number of nodes on the top level.

- **ScrollBarOptions**
  Reference to the scroll bar options class.

- **SearchBuffer**
  Current input string for incremental search.

- **Selected**
  Property to modify or determine the selection state of a node.

- **SelectedCount**
  Contains the number of selected nodes.

- **SelectionBlendFactor**
  Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

- **SelectionCurveRadius**
  Read or set the current corner radius for node selection rectangles.
- StateImages
  Reference to the images list which is used for the state images.
- TextMargin
  Read or set the distance of the node caption to its borders.
- TopNode
  The top node is the node which is currently at the top border of the client area.
- TotalCount
  Returns the number of nodes in the tree.
- TotalInternalDataSize
  Keeps the currently accumulated data size for one node.
- TreeOptions
  Reference to the tree's options.
- TreeStates
  Property which keeps a set of flags which indicate current operation and states of the tree.
- UpdateCount
  Not documented.
- VerticalAlignment
  Used to set a node's vertical button alignment with regard to the entire node rectangle.
- VisibleCount
  Number of currently visible nodes.
- VisiblePath
  Property to set or determine a node parent's expand states.
- WantTabs
  Read or set whether the tree wants to process tabs on its own.
Legend

TCustomVirtualDrawTree Class

Legend

- protected
- Event
- Method
- virtual
- public
- Property
- read only
TCustomVirtualStringTree Class Events

Events

- **OnGetHint**
  Virtual string tree event to query for a custom hint text.
- **OnGetText**
  Virtual string tree event to query for a node's normal or static text.
- **OnNewText**
  Virtual string tree event to pass edited text.
- **OnPaintText**
  Event to change text formatting for particular nodes.
- **OnShortenString**
  String tree event for custom handling of string abbreviations.

TBaseVirtualTree Class

- **OnAdvancedHeaderDraw**
  Header paint support event.
- **OnAfterCellPaint**
  Paint support event.
- **OnAfterItemErase**
  Paint support event.
- **OnAfterItemPaint**
  Paint support event.
- **OnAfterPaint**
  Paint support event.
- **OnBeforeCellPaint**
  Paint support event.
- **OnBeforeItemErase**
  Paint support event.
OnBeforeItemPaint
Paint support event.

OnBeforePaint
Paint support event.

OnChange
Navigation support event.

OnChecked
Check support event.

OnChecking
Check support event.

OnCollapsed
Miscellaneous event.

OnCollapsing
Miscellaneous event.

OnColumnClick
Header and column support event.

OnColumnDblClick
Header and column support event.

OnColumnResize
Header and column support routine.

OnCompareNodes
Sort and search support event.

OnCreateDataObject
Drag'n drop support event.

OnCreateDragManager
Drag'n drop support event.

OnCreateEditor
Editing support event.

OnDragAllowed
Drag'n drop support event.

OnDragDrop
Drag'n drop support event.

OnDragOver
Drag'n drop support event.

OnEditCancelled
Editing support event.
- **OnEdited**
  Editing support event.
- **OnEditing**
  Editing support event.
- **OnExpanded**
  Miscellaneous event.
- **OnExpanding**
  Miscellaneous event.
- **OnFocusChanged**
  Navigation support event.
- **OnFocusChanging**
  Navigation support event.
- **OnFreeNode**
  Data management node.
- **OnGetCellIsEmpty**
  Triggered when the tree control needs to know whether a given column is empty.
- **OnGetCursor**
  Miscellaneous event.
- **OnGetHeaderCursor**
  Header and column support event.
- **OnGetHelpContext**
  Miscellaneous event.
- **OnGetImageIndex**
  Display management event.
- **OnGetImageIndexEx**
  Not documented.
- **OnGetLineStyle**
  Display management event.
- **OnGetNodeDataSize**
  Data management event.
- **OnGetPopupMenu**
  Miscellaneous event.
- **OnGetUserClipboardFormats**
Drag'n drop and clipboard support event.

- **OnHeaderClick**
  Header & column support event.

- **OnHeaderDblClick**
  Header & column support event.

- **OnHeaderDragged**
  Header & column support event.

- **OnHeaderDraggedOut**
  Header & column support event.

- **OnHeaderDragging**
  Header & column support event.

- **OnHeaderDraw**
  Header & column support event.

- **OnHeaderDrawQueryElements**
  Header & column support event.

- **OnHeaderMouseDown**
  Header & column support event.

- **OnHeaderMouseMove**
  Header & column support event.

- **OnHeaderMouseUp**
  Header & column support event.

- **OnHotChange**
  Navigation support event.

- **OnIncrementalSearch**
  Miscellaneous event.

- **OnInitChildren**
  Node management event.

- **OnInitNode**
  Node management event.

- **OnKeyAction**
  Miscellaneous event.

- **OnLoadNode**
  Streaming support event.

- **OnMeasureItem**
  Miscellaneous event.
- **OnNodeCopied**
  Miscellaneous event.
- **OnNodeCopying**
  Miscellaneous event.
- **OnNodeMoved**
  Miscellaneous event.
- **OnNodeMoving**
  Miscellaneous event.
- **OnPaintBackground**
  Paint support event.
- **OnRenderOLEData**
  Drag'n drop and clipboard support event.
- **OnResetNode**
  Node management event.
- **OnSaveNode**
  Streaming support event.
- **OnScroll**
  Miscellaneous event.
- **OnShowScrollbar**
  Not documented.
- **OnStateChange**
  Miscellaneous event.
- **OnStructureChange**
  Miscellaneous event.
- **OnUpdating**
  Miscellaneous event.
TCustomVirtualStringTree Class Methods

Methods

- **AdjustPaintCellRect**
  Method which can be used by descentants to adjust the given rectangle during a paint cycle.

- **CalculateTextWidth**
  Not documented.

- **ColumnIsEmpty**
  Used to determine if a cell is considered as being empty.

- **ComputeNodeHeight**
  Not documented.

- **ContentToClipboard**
  Not documented.

- **ContentToHTML**
  Not documented.

- **ContentToRTF**
  Not documented.

- **ContentToText**
  Not documented.

- **ContentToUnicode**
  Not documented.

- **Create**
  Constructor of the control

- **DefineProperties**
  Helper method to customize loading and saving persistent tree data.

- **DoCreateEditor**
  Not documented.

- **DoGetNodeHint**
  Not documented.
- **DoGetNodeTooltip**
  Not documented.

- **DoGetNodeWidth**
  Overridable method which always returns 0.

- **DoGetText**
  Not documented.

- **DoIncrementalSearch**
  Not documented.

- **DoNewText**
  Not documented.

- **DoPaintNode**
  Overridable method which does nothing.

- **DoPaintText**
  Not documented.

- **DoShortenString**
  Not documented.

- **DoTextDrawing**
  Not documented.

- **DoTextMeasuring**
  Not documented.

- **GetOptionsClass**
  Customization helper to determine which options class the tree should use.

- **GetTextInfo**
  Helper method for node editors, hints etc.

- **InternalData**
  Returns the address of the internal data for a tree class.

- **InvalidateNode**
  Invalidates the given node.

- **MainColumnChanged**
  Not documented.

- **Path**
  Not documented.

- **ReadChunk**
  Not documented.
ReadOldStringOptions
Not documented.

ReinitNode
Forces a reinitialization of the given node.

RenderOLEData
Renders pending OLE data.

WriteChunks
Writes the core chunks for the given node to the given stream.

TBaseVirtualTree Class

AbsoluteIndex
Reads the overall index of a node.

AddChild
Creates and adds a new child node to given node.

AddFromStream
Adds the content from the given stream to the given node.

AddToSelection
Adds one or more nodes to the current selection.

AdjustPaintCellRect
Used in descendent to modify the clip rectangle of the current column while painting a certain node.

AdjustPanningCursor
Loads the proper cursor which indicates into which direction scrolling is done.

AdviseChangeEvent
Used to register a delayed change event.

AllocateInternalDataArea
Registration method to allocate tree internal data per node.

Animate
Support method for animated actions in the tree view.

Assign
Used to copy properties from another Virtual Treeview.

BeginDrag
Starts an OLE drag’n drop operation.
Adds the tree into a special synchronized mode.

**BeginUpdate**
Locks the tree view to perform several update operations.

**CalculateSelectionRect**
Support method for draw selection.

**CanAutoScroll**
Determines whether the tree can currently auto scroll its window.

**CancelCutOrCopy**
Cancels any pending cut or copy clipboard operation.

**CancelEditNode**
Cancel the current edit operation, if there is any.

**CanEdit**
Determines whether a node can be edited or not.

**CanFocus**
Support method to determine whether the tree window can receive the input focus.

**CanShowDragImage**
Determines whether a drag image should be shown.

**Change**
Central method called when a node's selection state changes.

**ChangeScale**
Helper method called by the VCL when control resizing is due.

**CheckParentCheckState**
Helper method for recursive check state changes.

**Clear**
Clears the tree and removes all nodes.

**ClearChecked**
Not documented.

**ClearSelection**
Removes all nodes from the current selection.

**ClearTempCache**
Helper method to clear the internal temporary node cache.

**ColumnIsEmpty**
Used to determine if a cell is considered as being empty.
CopyTo
Copies **Source** and all its child nodes to **Target**.

CopyToClipboard
Copies all currently selected nodes to the clipboard.

CountLevelDifference
Determines the level difference of two nodes.

CountVisibleChildren
Determines the number of visible child nodes of the given node.

Create
Constructor of the control

CreateParams
Prepares the creation of the controls window handle.

CreateWnd
Initializes data, which depends on the window handle.

CutToClipboard
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties
Helper method to customize loading and saving persistent tree data.

DeleteChildren
Removes all child nodes from the given node.

DeleteNode
Removes the given node from the tree.

DeleteSelectedNodes
Removes all currently selected nodes from the tree.

Destroy
Destructor of the control.

DetermineHiddenChildrenFlag
Determines whether all children of a given node are hidden.

DetermineHiddenChildrenFlagAllNodes
Determines whether all children of all nodes are hidden.

DetermineHitPositionLTR
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineHitPositionRTL
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineNextCheckState
Not documented.

DetermineScrollDirections
Not documented.

DoAdvancedHeaderDraw
Not documented.

DoAfterCellPaint
Not documented.

DoAfterItemErase
Not documented.

DoAfterItemPaint
Not documented.

DoAfterPaint
Not documented.

DoAutoScroll
Enables or disables the auto scroll timer.

DoBeforeCellPaint
Not documented.

DoBeforeDrag
Not documented.

DoBeforeItemErase
Not documented.

DoBeforeItemPaint
Not documented.

DoBeforePaint
Not documented.

DoCancelEdit
Called when the tree should stop editing without accepting changed values.

DoCanEdit
Not documented.

DoChange
Not documented.
DoCheckClick
Not documented.
DoChecked
Not documented.
DoChecking
Not documented.
DoCollapsed
Not documented.
DoCollapsing
Not documented.
DoColumnClick
Not documented.
DoColumnDblClick
Not documented.
DoColumnResize
Not documented.
DoCompare
Not documented.
DoCreateDataObject
Not documented.
DoCreateDragManager
Not documented.
DoCreateEditor
Not documented.
DoDragDrop
Not documented.
DoDragExpand
Not documented.
DoDragging
Internal method which handles drag' drop.
DoDragOver
Not documented.
DoEdit
Initiates editing of the currently set focused column and edit node.
DoEndDrag
Not documented.

**:DoEndEdit**
Stops the current edit operation and takes over the new content.

**:DoExpanded**
Not documented.

**:DoExpanding**
Not documented.

**:DoFocusChange**
Not documented.

**:DoFocusChanging**
Not documented.

**:DoFocusNode**
Internal method to set the focused node.

**:DoFreeNode**
Not documented.

**:DoGetAnimationType**
Determines the type of animation to be used.

**:DoGetCursor**
Not documented.

**:DoGetHeaderCursor**
Not documented.

**:DoGetImageIndex**
Not documented.

**:DoGetLineStyle**
Not documented.

**:DoGetNodeHint**
Not documented.

**:DoGetNodeTooltip**
Not documented.

**:DoGetNodeWidth**
Overridable method which always retuns 0.

**:DoGetPopupMenu**
Overridable method which triggers the OnGetPopup event.

**:DoGetUserClipboardFormats**
Not documented.
DoHeaderClick
Not documented.
DoHeaderDblClick
Not documented.
DoHeaderDragged
Not documented.
DoHeaderDraggedOut
Not documented.
DoHeaderDragging
Not documented.
DoHeaderDraw
Not documented.
DoHeaderDrawQueryElements
Not documented.
DoHeaderMouseDown
Not documented.
DoHeaderMouseMove
Not documented.
DoHeaderMouseUp
Not documented.
DoHotChange
Not documented.
DoIncrementalSearch
Not documented.
DoInitChildren
Not documented.
DoInitNode
Not documented.
DoKeyAction
Not documented.
DoLoadUserData
Not documented.
DoMeasureItem
Not documented.
DoNodeCopied

Not documented.

- **DoNodeCopying**
  Not documented.

- **DoNodeMoved**
  Not documented.

- **DoNodeMoving**
  Not documented.

- **DoPaintBackground**
  Not documented.

- **DoPaintDropMark**
  Overridable method which draws the small line on top of a node's image depending on the current drop state.

- **DoPaintNode**
  Overridable method which does nothing.

- **DoPopupMenu**
  Overridable method which shows the popup menu for the given node.

- **DoRenderOLEData**
  Not documented.

- **DoReset**
  Not documented.

- **DoSaveUserData**
  Not documented.

- **DoScroll**
  Overridable method which triggers the `OnScroll` event.

- **DoSetOffsetXY**
  Internal core routine to set the tree's scroll position.

- **DoShowScrollbar**
  Not documented.

- **DoStartDrag**
  Not documented.

- **DoStateChange**
  Not documented.

- **DoStructureChange**
  Not documented.

- **DoTimerScroll**
  Not documented.
Callback method which is triggered whenever the scroll timer fires.

- **DoUpdating**
  Not documented.

- **DoValidateCache**
  Not documented.

- **DragCanceled**
  Called by the VCL when a drag'n drop operation was canceled by the user.

- **DragDrop**
  Helper method, which is used when a drag operation is finished.

- **DragEnter**
  Not documented.

- **DragFinished**
  Called when a drag operation is finished (accepted or cancelled).

- **Dragging**
  Returns true if a drag'n drop operation is in progress.

- **DragLeave**
  Not documented.

- **DragOver**
  Not documented.

- **DrawDottedHLine**
  Not documented.

- **DrawDottedVLine**
  Not documented.

- **EditNode**
  Starts editing the given node if allowed to.

- **EndEditNode**
  Stops node editing if it was started before.

- **EndSynch**
  Counterpart to **BeginSynch**.

- **EndUpdate**
  Resets the update lock set by **BeginUpdate**.

- **ExecuteAction**
  Not documented.

- **FindNodeInSelection**
Helper method to find the given node in the current selection.

FinishChunkHeader
Not documented.

FinishCutOrCopy
Stops any pending cut or copy clipboard operation.

FlushClipboard
Renders all pending clipboard data.

FontChanged
Not documented.

FullCollapse
Collapses all nodes in the tree.

FullExpand
Expands all nodes in the tree.

GetBorderDimensions
Not documented.

GetCheckImage
Not documented.

GetCheckImageListFor
Not documented.

GetColumnClass
Returns the class to be used to manage columns in the tree.

GetControlsAlignment
Not documented.

GetDisplayRect
Returns the visible region used by the given node in client coordinates.

GetFirst
Group of node navigation functions.

GetFirstChild
Group of node navigation functions.

GetFirstCutCopy
Group of node navigation functions.

GetFirstInitialized
Group of node navigation functions.

- **GetFirstNoInit**
  Group of node navigation functions.

- **GetFirstSelected**
  Group of node navigation functions.

- **GetFirstVisible**
  Group of node navigation functions.

- **GetFirstVisibleChild**
  Group of node navigation functions.

- **GetFirstVisibleChildNoInit**
  Group of node navigation functions.

- **GetFirstVisibleNoInit**
  Group of node navigation functions.

- **GetHeaderClass**
  Returns the header class to be used by the tree.

- **GetHintWindowClass**
  Not documented.

- **GetHitTestInfoAt**
  Returns information about the node at the given position.

- **GetImageIndex**
  Not documented.

- **GetLast**
  Group of node navigation functions.

- **GetLastChild**
  Group of node navigation functions.

- **GetLastChildNoInit**
  Group of node navigation functions.

- **GetLastInitialized**
  Group of node navigation functions.

- **GetLastNoInit**
  Group of node navigation functions.

- **GetLastVisible**
  Group of node navigation functions.

- **GetLastVisibleChild**
  Group of node navigation functions.
GetLastVisibleChildNoInit
Group of node navigation functions.

GetLastVisibleNoInit
Group of node navigation functions.

GetMaxColumnWidth
Returns the width of the largest node in the given column.

GetMaxRightExtend
Determines the maximum width of the currently visible part of the tree.

GetNativeClipboardFormats
Used to let descendants and the application add their own supported clipboard formats.

GetNext
Group of node navigation functions.

GetNextChecked
Not documented.

GetNextCutCopy
Group of node navigation functions.

GetNextInitialized
Group of node navigation functions.

GetNextNoInit
Group of node navigation functions.

GetNextSelected
Group of node navigation functions.

GetNextSibling
Group of node navigation functions.

GetNextVisible
Group of node navigation functions.

GetNextVisibleNoInit
Group of node navigation functions.

GetNextVisibleSibling
Group of node navigation functions.

GetNextVisibleSiblingNoInit
Group of node navigation functions.

GetNodeAt
Not documented.
GetNodeData
Returns the address of the user data area of the given node.

GetNodeLevel
Returns the indentation level of the given node.

GetOptionsClass
Customization helper to determine which options class the tree should use.

GetPrevious
Group of node navigation functions.

GetPreviousInitialized
Group of node navigation functions.

GetPreviousNoInit
Group of node navigation functions.

GetPreviousSibling
Group of node navigation functions.

GetPreviousVisible
Group of node navigation functions.

GetPreviousVisibleNoInit
Group of node navigation functions.

GetPreviousVisibleSibling
Group of node navigation functions.

GetPreviousVisibleSiblingNoInit
Group of node navigation functions.

GetSortedCutCopySet
Returns a sorted list of nodes, which are marked for cut or copy clipboard operation.

GetSortedSelection
Returns a sorted list of all currently selected nodes.

GetTextInfo
Helper method for node editors, hints etc.

GetTreeFromDataObject
OLE drag'n drop and clipboard support method.

GetTreeRect
Returns the size of the virtual tree image.

GetVisibleParent
Returns the first (nearest) parent node, which is visible.

- **HandleHotTrack**
  Not documented.

- **HandleIncrementalSearch**
  Not documented.

- **HandleMouseDb1Click**
  Not documented.

- **HandleMouseDown**
  Not documented.

- **HandleMouseUp**
  Not documented.

- **HasAsParent**
  Determines if the given node has got another node as one of its parents.

- **HasImage**
  Not documented.

- **HasPopupMenu**
  Determines whether there is a pop up menu assigned to the tree.

- **InitChildren**
  Not documented.

- **InitNode**
  Not documented.

- **InsertNode**
  Inserts a new node and returns it to the caller.

- **InternalAddFromStream**
  Not documented.

- **InternalAddToSelection**
  Not documented.

- **InternalCacheNode**
  Not documented.

- **InternalClearSelection**
  Not documented.

- **InternalConnectNode**
  Not documented.

- **InternalData**
Returns the address of the internal data for a tree class.

- **InternalDisconnectNode**
  Not documented.

- **InternalRemoveFromSelection**
  Not documented.

- **InvalidateCache**
  Empties the internal node cache and marks it as invalid.

- **InvalidateChildren**
  Invalidates all children of the given node.

- **InvalidateColumn**
  Invalidates the client area part of a column.

- **InvalidateNode**
  Invalidates the given node.

- **InvalidateToBottom**
  Invalidates the client area starting with the top position of the given node.

- **InvertSelection**
  Inverts the current selection.

- **IsEditing**
  Tells the caller whether the tree is currently in edit mode.

- **IsMouseSelecting**
  Tell the caller whether the tree is currently in draw selection mode.

- **IterateSubtree**
  Iterator method to go through all nodes of a given sub tree.

- **Loaded**
  Not documented.

- **LoadFromFile**
  Loads previously streamed out tree data back in again.

- **LoadFromStream**
  Loads previously streamed out tree data back in again.

- **MainColumnChanged**
  Not documented.

- **MarkCutCopyNodes**
  Not documented.

- **MeasureItemHeight**
Not documented.

MouseMove
Not documented.

MoveTo
Moves **Source** and all its child nodes to **Target**.

Notification
Not documented.

OriginalWMNCNPaint
Not documented.

Paint
TControl's Paint method used here to display the tree.

PaintCheckImage
Not documented.

PaintImage
Not documented.

PaintNodeButton
Not documented.

PaintSelectionRectangle
Not documented.

PaintTree
Main paint routine for the tree image.

PaintTreeLines
Not documented.

PanningWindowProc
Not documented.

PasteFromClipboard
Inserts the content of the clipboard into the tree.

PrepareDragImage
Not documented.

Print
Not documented.

ProcessDrop
Helper method to ease OLE drag'n drop operations.

ProcessOLEData
Takes serialized OLE tree data and reconstructs the former structure.
ReadChunk
Not documented.
ReadNode
Not documented.
RedirectFontChangeEvent
Not documented.
ReinitChildren
Forces all child nodes of Node to be reinitialized.
ReinitNode
Forces a reinitialization of the given node.
RemoveFromSelection
Removes the given node from the current selection.
RenderOLEData
Renders pending OLE data.
RepaintNode
Causes the treeview to repaint the given node.
ResetNode
Resets the given node to uninitialized.
ResetRangeAnchor
Not documented.
RestoreFontChangeEvent
Not documented.
SaveToFile
Saves the entire content of the tree into a file or stream.
SaveToStream
Saves the entire content of the tree into a file or stream.
ScrollIntoView
Scrolls the tree so that the given node comes in the client area.
SelectAll
Selects all nodes in the tree.
SelectNodes
Selects a range of nodes.
SetBiDiMode
Not documented.
SetFocusedNodeAndColumn
Not documented.

SkipNode
Not documented.

Sort
Sorts the given node.

SortTree
Sorts the entire tree view.

StartWheelPanning
Not documented.

StopWheelPanning
Not documented.

StructureChange
Not documented.

SuggestDropEffect
Not documented.

ToggleNode
Changes a node's expand state to the opposite state.

ToggleSelection
Toggles the selection state of a range of nodes.

UnselectNodes
Deselects a range of nodes.

UpdateAction
Not documented.

UpdateDesigner
Not documented.

UpdateEditBounds
Not documented.

UpdateHeaderRect
Not documented.

UpdateHorizontalScrollBar
Applies changes to the horizontal and vertical scrollbars.

UpdateScrollBars
Applies changes to the horizontal and vertical scrollbars.

UpdateVerticalScrollBar
Applies changes to the horizontal and vertical scrollbars.
- **UpdateWindowAndDragImage**
  Not documented.
- **UseRightToLeftReading**
  Helper method for right-to-left layout.
- **ValidateCache**
  Initiates the validation of the internal node cache.
- **ValidateChildren**
  Validates all children of a given node.
- **ValidateNode**
  Validates a given node.
- **ValidateNodeDataSize**
  Helper method for node data size initialization.
- **WndProc**
  Redirected window procedure to do some special processing.
- **WriteChunks**
  Writes the core chunks for the given node to the given stream.
- **WriteNode**
  Writes the cover (envelop) chunk for the given node to the given stream.
**TCustomVirtualStringTree Class Properties**

**Properties**

- **DefaultText**
  Not documented.
- **EllipsisWidth**
  Not documented.
- **Text**
  Not documented.
- **TreeOptions**
  Reference to the tree's options.

**TBaseVirtualTree Class**

- **Alignment**
  Determines the horizontal alignment of text if no columns are defined.
- **AnimationDuration**
  Determines the maximum duration the tree can use to play an animation.
- **AutoExpandDelay**
  Time delay after which a node gets expanded if it is the current drop target.
- **AutoScrollDelay**
  Time which determines when auto scrolling should start.
- **AutoScrollInterval**
  Time interval between scroll events when doing auto scroll.
- **Background**
  Holds a background image for the tree.
- **BackgroundOffsetX**
  Horizontal offset of the background image.
BackgroundOffsetY
   Vertical offset of the background image.

BorderStyle
   Same as TForm.BorderStyle.

ButtonFillMode
   Determines how to fill the background of the node buttons.

ButtonStyle
   Determines the look of node buttons.

ChangeDelay
   Time which determines when the OnChange event should be triggered after the actual change event.

CheckImageKind
   Determines which images should be used for checkboxes and radio buttons.

CheckImages
   Not documented.

CheckState
   Read or set the check state of a node.

CheckType
   Read or set the check type of a node.

ChildCount
   Read or set the number of child nodes of a node.

ChildrenInitialized
   Read whether a node's child count has been initialized already.

ClipboardFormats
   Special class to keep a list of clipboard format descriptions.

Colors
   A collection of colors used in the tree.

CustomCheckImages
   Assign your own image list to get the check images you like most.

DefaultNodeHeight
   Read or set the height new nodes get as initial value.

DefaultPasteMode
   Read or set the value, which determines where to add pasted nodes to.
DragHeight
Read or set the vertical limit of the internal drag image.

DragImage
Holds the instance of the internal drag image.

DragImageKind
Read or set what should be shown in the drag image.

DragManager
Holds the reference to the internal drag manager.

DragOperations
Read or set which drag operations may be allowed in the tree.

DragSelection
Keeps a temporary list of nodes during drag'n drop.

DragType
Read or set which subsystem should be used for dragging.

DragWidth
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode
Read or set how multiselection with the mouse is to be visualized.

DropTargetNode
Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

EditColumn
Not documented.

EditDelay
Read or set the maximum time between two single clicks on the same node, which should start node editing.

EditLink
Keeps a reference to the internal edit link during a node edit operation.

Expanded
Read or set the expanded state of a particular node.

FocusedColumn
Read or set the currently focused column.

FocusedNode
Read or set the currently focused node.
Font
Same as TWinControl.Font.

FullyVisible
Read or set whether a node is fully visible or not.

HasChildren
Read or set whether a node has got children.

Header
Provides access to the header instance.

HeaderRect
Returns the non-client-area rectangle used for the header.

HintAnimation
Read or set the current hint animation type.

HintMode
Read or set what type of hint you want for the tree view.

HotCursor
Read or set which cursor should be used for hot nodes.

HotNode
Read, which node is currently the hot node.

Images
Read or set the tree's normal image list.

IncrementalSearch
Read or set the current incremental search mode.

IncrementalSearchDirection
Read or set the direction to be used for incremental search.

IncrementalSearchStart
Read or set where to start incremental search.

IncrementalSearchTimeout
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

Indent
Read or set the indentation amount for node levels.

IsDisabled
Read or set the enabled state of the given node.

IsVisible
Read or set the visibility state of the given node.
- LastClickPos
  Used for retained drag start and wheel mouse scrolling.
- LastDropMode
  Read how the last drop operation finished.
- LineMode
  Read or set the mode of the tree lines.
- LineStyle
  Read or set the mode of the tree lines.
- Margin
  Read or set the tree's node margin.
- MultiLine
  Read or toggle the multiline feature for a given node.
- NodeAlignment
  Read or set the node alignment value.
- NodeDataSize
  Read or set the extra data size for each node.
- NodeHeight
  Read or set a node's height.
- NodeParent
  Read or set a node's parent node.
- OffsetX
  Read or set the tree's current horizontal and vertical scroll offsets.
- OffsetXY
  Read or set the tree's current horizontal and vertical scroll offsets.
- OffsetY
  Read or set the tree's current horizontal and vertical scroll offsets.
- RootNode
  Reference to the internal root node which is the anchor of the entire tree node hierarchy.
- RootNodeCount
  Read or set the number of nodes on the top level.
- ScrollBarOptions
  Reference to the scroll bar options class.
- SearchBuffer
  Current input string for incremental search.
Selected
Property to modify or determine the selection state of a node.

SelectedCount
Contains the number of selected nodes.

SelectionBlendFactor
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

SelectionCurveRadius
Read or set the current corner radius for node selection rectangles.

StateImages
Reference to the images list which is used for the state images.

TextMargin
Read or set the distance of the node caption to its borders.

TopNode
The top node is the node which is currently at the top border of the client area.

TotalCount
Returns the number of nodes in the tree.

TotalInternalDataSize
Keeps the currently accumulated data size for one node.

TreeOptions
Reference to the tree's options.

TreeStates
Property which keeps a set of flags which indicate current operation and states of the tree.

UpdateCount
Not documented.

VerticalAlignment
Used to set a node's vertical button alignment with regard to the entire node rectangle.

VisibleCount
Number of currently visible nodes.

VisiblePath
Property to set or determine a node parent's expand states.

WantTabs
Read or set whether the tree wants to process tabs on its own.
See Also

TCustomVirtualStringTree.OnGetText Event

See Also

OnPaintText
See Also
TCustomVirtualStringTree.OnNewText Event

See Also
OnCreateEditor, OnEdited
See Also
TCustomVirtualStringTree.OnPaintText Event

See Also
Paint cycles and stages
See Also
TCustomVirtualStringTree.ColumnIsEmpty Method

See Also
toAutoSpanColumns
See Also
TCustomVirtualStringTree.InternalData Method

See Also
Data handling
See Also
TCustomVirtualStringTree.WriteChunks Method

See Also
WriteNode, SaveToStream
TCustomVirtualTreeOptions Class Methods

TCustomVirtualTreeOptions Class | Legend

Methods

- 🌍.AssignTo
  - Used to copy this option class to another option collection.
- 🌍.Create
  - Constructor of the class.
TCustomVirtualTreeOptions Class Properties

TCustomVirtualTreeOptions Class | Legend

Properties

- AnimationOptions
  Options related to animations.
- AutoOptions
  Options related to automatic actions.
- MiscOptions
  Options not related to any other category.
- Owner
  Owner tree to which the property class belongs.
- PaintOptions
  Options related to painting.
- SelectionOptions
  Options related to the way nodes can be selected.
Legend

TCustomVirtualTreeOptions Class

Legend

● protected
● Property
● public
● read only
● Method
● virtual
TEnumFormatEtc Class Methods

Methods

- **Clone**
  Not documented.

- **Create**
  Not documented.

- **Next**
  Not documented.

- **Reset**
  Not documented.

- **Skip**
  Not documented.
Legend
  TEnumFormatEtc Class

Legend

public

Method
TScrollBarOptions Class Methods

TScrollBarOptions Class | Legend

Methods

- Assign
  Not documented.
- Create
  Not documented.
- GetOwner
  Not documented.
# TScrollBarOptions Class Properties

TScrollBarOptions Class | Legend

## Properties

- **AlwaysVisible**
  - Not documented.
- **HorizontalIncrement**
  - Not documented.
- **ScrollBars**
  - Not documented.
- **ScrollBarStyle**
  - Not documented.
- **VerticalIncrement**
  - Not documented.
TStringEditLink Class Methods

Methods

- **BeginEdit**
  This function will be called by the virtual string tree when the editing starts.

- **CancelEdit**
  This function will be called by the virtual string tree when the current editing is about to be cancelled.

- **Create**
  Constructor of the class.

- **Destroy**
  Destructor of the class.

- **EndEdit**
  This function will be called by the virtual string tree when the current editing is being finished.

- **GetBounds**
  The virtual string tree uses this function to get the current bounding rect of the node editor.

- **PrepareEdit**
  This function is called by a virtual string tree to initialize the node editor.

- **ProcessMessage**
  This function is used to forward messages being directed to the virtual string tree.

- **SetBounds**
  The virtual string tree calls this function to initialize the bounding rect of the node editor.

IVTEditLink Interface
BeginEdit
This function will be called by the virtual tree when the editing starts.

CancelEdit
This function will be called by the virtual tree when the current editing is about to be cancelled.

EndEdit
This function will be called by the virtual tree when the current editing is being finished.

GetBounds
The virtual tree can use this function to get the current bounding rect of the node editor.

PrepareEdit
This function is called by a virtual tree to initialize the node editor.

ProcessMessage
This function is used to forward messages being directed to the virtual tree.

SetBounds
The virtual tree calls this function to initialize the bounding rectangle of the node editor.
TStringEditLink Class Properties

Properties

Edit
Not documented.
Legend

TStringEditLink Class

Legend

- public
- Property
- Method
- virtual
**TStringTreeOptions Class Methods**

**Methods**

**TCustomStringTreeOptions Class**

- **AssignTo**
  Used to copy the options class.
- **Create**
  The constructor of the class.

**TCustomVirtualTreeOptions Class**

- **AssignTo**
  Used to copy this option class to another option collection.
- **Create**
  Constructor of the class.
TStringTreeOptions Class Properties

Properties

- **AnimationOptions**
  Options related to animations.
- **AutoOptions**
  Options related to automatic actions.
- **MiscOptions**
  Options not related to any other category.
- **PaintOptions**
  Options related to painting.
- **SelectionOptions**
  Options related to the way nodes can be selected.
- **StringUtil**
  The new options introduced by the class.

TCustomStringTreeOptions Class

- **StringOptions**
  The new options introduced by the class.

TCustomVirtualTreeOptions Class

- **AnimationOptions**
  Options related to animations.
- **AutoOptions**
  Options related to automatic actions.
- **MiscOptions**
  Options not related to any other category.
- **Owner**
Owner tree to which the property class belongs.

- **PaintOptions**
  Options related to painting.

- **SelectionOptions**
  Options related to the way nodes can be selected.
Legend
TStringTreeOptions Class

Legend

- published
- Property
- protected
- public
- read only
- Method
- virtual
TVirtualDrawTree Class Events

Events

- **OnAdvancedHeaderDraw**
  Header paint support event.

- **OnAfterCellPaint**
  Paint support event.

- **OnAfterItemErase**
  Paint support event.

- **OnAfterItemPaint**
  Paint support event.

- **OnAfterPaint**
  Paint support event.

- **OnBeforeCellPaint**
  Paint support event.

- **OnBeforeItemErase**
  Paint support event.

- **OnBeforeItemPaint**
  Paint support event.

- **OnBeforePaint**
  Paint support event.

- **OnChange**
  Navigation support event.

- **OnChecked**
  Check support event.

- **OnChecking**
  Check support event.

- **OnCollapsed**
  Miscellaneous event.
OnCollapsing
Miscellaneous event.

OnColumnClick
Header and column support event.

OnColumnDb1Click
Header and column support event.

OnColumnResize
Header and column support routine.

OnCompareNodes
Sort and search support event.

OnCreateDataObject
Drag'n drop support event.

OnCreateDragManager
Drag'n drop support event.

OnCreateEditor
Editing support event.

OnDragAllowed
Drag'n drop support event.

OnDragDrop
Drag'n drop support event.

OnDragOver
Drag'n drop support event.

OnDrawHint
Triggered when a node hint or tooltip must be drawn.

OnDrawNode
Triggered when a node must be drawn.

OnEdited
Editing support event.

OnEditing
Editing support event.

OnExpanded
Miscellaneous event.

OnExpanding
Miscellaneous event.

OnFocusChanged
Navigation support event.

- **OnFocusChanging**
  Navigation support event.

- **OnFreeNode**
  Data management node.

- **OnGetCellIsEmpty**
  Triggered when the tree control needs to know whether a given column is empty.

- **OnGetCursor**
  Miscellaneous event.

- **OnGetHeaderCursor**
  Header and column support event.

- **OnGetHelpContext**
  Miscellaneous event.

- **OnGetHintSize**
  Triggered when a node hint or tooltip is about to show.

- **OnGetImageIndex**
  Display management event.

- **OnGetImageIndexEx**
  Not documented.

- **OnGetLineStyle**
  Display management event.

- **OnGetNodeDataSize**
  Data management event.

- **OnGetNodeWidth**
  Triggered when a node is about to be drawn.

- **OnGetPopupMenu**
  Miscellaneous event.

- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.

- **OnHeaderClick**
  Header & column support event.

- **OnHeaderDbClick**
  Header & column support event.

- **OnHeaderDragged**
Header & column support event.

- OnHeaderDraggedOut
- OnHeaderDragging
- OnHeaderDraw
- OnHeaderMouseDown
- OnHeaderMouseMove
- OnHeaderMouseUp
- OnHotChange

Navigation support event.

- OnIncrementalSearch

Miscellaneous event.

- OnInitChildren
- OnInitNode
- OnKeyAction
- OnLoadNode
- OnMeasureItem
- OnNodeCopied
- OnNodeCopying
- OnNodeMoved
OnNodeMoving
Miscellaneous event.

OnPaintBackground
Paint support event.

OnRenderOLEData
Drag'n drop and clipboard support event.

OnResetNode
Node management event.

OnSaveNode
Streaming support event.

OnScroll
Miscellaneous event.

OnShowScrollbar
Not documented.

OnStateChange
Miscellaneous event.

OnStructureChange
Miscellaneous event.

OnUpdating
Miscellaneous event.

TCustomVirtualDrawTree Class

OnDrawHint
Triggered when a node hint or tooltip must be drawn.

OnDrawNode
Triggered when a node must be drawn.

OnGetHintSize
Triggered when a node hint or tooltip is about to show.

OnGetNodeWidth
Triggered when a node is about to be drawn.

TBaseVirtualTree Class

OnAdvancedHeaderDraw
Header paint support event.
OnAfterCellPaint
Paint support event.

OnAfterItemErase
Paint support event.

OnAfterItemPaint
Paint support event.

OnAfterPaint
Paint support event.

OnBeforeCellPaint
Paint support event.

OnBeforeItemErase
Paint support event.

OnBeforeItemPaint
Paint support event.

OnBeforePaint
Paint support event.

OnChange
Navigation support event.

OnChecked
Check support event.

OnChecking
Check support event.

OnCollapsible
Miscellaneous event.

OnCollapsing
Miscellaneous event.

OnColumnClick
Header and column support event.

OnColumnDblClick
Header and column support event.

OnColumnResize
Header and column support routine.

OnCompareNodes
Sort and search support event.

OnCreateDataObject
Drag'n drop support event.

- **OnCreateDragManager**
- **OnCreateEditor**
- **OnDragAllowed**
- **OnDragDrop**
- **OnDragOver**
- **OnEditCancelled**
- **OnEdited**
- **OnEditing**
- **OnExpanded**
- **OnExpanding**
- **OnFocusChanged**
- **OnFocusChanging**
- **OnFreeNode**
- **OnGetCellIsEmpty**
- **OnGetCursor**
- **OnGetHeaderCursor**
- **OnGetHelpContext**

Triggered when the tree control needs to know whether a given column is empty.
Miscellaneous event.

- **OnGetImageIndex**
  Display management event.

- **OnGetImageIndexEx**
  Not documented.

- **OnGetLineStyle**
  Display management event.

- **OnGetNodeDataSize**
  Data management event.

- **OnGetPopupMenu**
  Miscellaneous event.

- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.

- **OnHeaderClick**
  Header & column support event.

- **OnHeaderDbClick**
  Header & column support event.

- **OnHeaderDragged**
  Header & column support event.

- **OnHeaderDraggedOut**
  Header & column support event.

- **OnHeaderDragging**
  Header & column support event.

- **OnHeaderDraw**
  Header & column support event.

- **OnHeaderDrawQueryElements**
  Header & column support event.

- **OnHeaderMouseDown**
  Header & column support event.

- **OnHeaderMouseMove**
  Header & column support event.

- **OnHeaderMouseUp**
  Header & column support event.

- **OnHotChange**
  Navigation support event.
OnIncrementalSearch
  Miscellaneous event.

OnInitChildren
  Node management event.

OnInitNode
  Node management event.

OnKeyAction
  Miscellaneous event.

OnLoadNode
  Streaming support event.

OnMeasureItem
  Miscellaneous event.

OnNodeCopied
  Miscellaneous event.

OnNodeCopying
  Miscellaneous event.

OnNodeMoved
  Miscellaneous event.

OnNodeMoving
  Miscellaneous event.

OnPaintBackground
  Paint support event.

OnRenderOLEData
  Drag'n drop and clipboard support event.

OnResetNode
  Node management event.

OnSaveNode
  Streaming support event.

OnScroll
  Miscellaneous event.

OnShowScrollbar
  Not documented.

OnStateChange
  Miscellaneous event.

OnStructureChange
Miscellaneous event.

OnUpdating

Miscellaneous event.
**TVirtualDrawTree Class Methods**

**TVirtualDrawTree Class** | **Legend**

### Methods

- **GetOptionsClass**
  Customization helper to determine which options class the tree should use.

**TCustomVirtualDrawTree Class**

- **DoDrawHint**
  Overridable method which triggers OnDrawHint.
- **DoGetHintSize**
  Overridable method which triggers OnGetHintSize.
- **DoGetNodeWidth**
  Overridable method which triggers OnGetNodeWidth.
- **DoPaintNode**
  Overridable method which triggers OnPaintNode.

**TBaseVirtualTree Class**

- **AbsoluteIndex**
  Reads the overall index of a node.
- **AddChild**
  Creates and adds a new child node to given node.
- **AddFromStream**
  Adds the content from the given stream to the given node.
- **AddToSelection**
  Adds one or more nodes to the current selection.
- **AdjustPaintCellRect**
  Used in descendents to modify the clip rectangle of the current column
while painting a certain node.

- **AdjustPanningCursor**
  Loads the proper cursor which indicates into which direction scrolling is done.

- **AdviseChangeEvent**
  Used to register a delayed change event.

- **AllocateInternalDataArea**
  Registration method to allocate tree internal data per node.

- **Animate**
  Support method for animated actions in the tree view.

- **Assign**
  Used to copy properties from another Virtual Treeview.

- **BeginDrag**
  Starts an OLE drag'n drop operation.

- **BeginSynch**
  Enters the tree into a special synchronized mode.

- **BeginUpdate**
  Locks the tree view to perform several update operations.

- **CalculateSelectionRect**
  Support method for draw selection.

- **CanAutoScroll**
  Determines whether the tree can currently auto scroll its window.

- **CancelCutOrCopy**
  Canceles any pending cut or copy clipboard operation.

- **CancelEditNode**
  Cancel the current edit operation, if there is any.

- **CanEdit**
  Determines whether a node can be edited or not.

- **CanFocus**
  Support method to determine whether the tree window can receive the input focus.

- **CanShowDragImage**
  Determines whether a drag image should be shown.

- **Change**
  Central method called when a node's selection state changes.
ChangeScale
   Helper method called by the VCL when control resizing is due.

CheckParentCheckState
   Helper method for recursive check state changes.

Clear
   Clears the tree and removes all nodes.

ClearChecked
   Not documented.

ClearSelection
   Removes all nodes from the current selection.

ClearTempCache
   Helper method to clear the internal temporary node cache.

ColumnIsEmpty
   Used to determine if a cell is considered as being empty.

CopyTo
   Copies Source and all its child nodes to Target.

CopyToClipBoard
   Copies all currently selected nodes to the clipboard.

CountLevelDifference
   Determines the level difference of two nodes.

CountVisibleChildren
   Determines the number of visible child nodes of the given node.

Create
   Constructor of the control

CreateParams
   Prepares the creation of the controls window handle.

CreateWnd
   Initializes data, which depends on the window handle.

CutToClipBoard
   Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties
   Helper method to customize loading and saving persistent tree data.

DeleteChildren
   Removes all child nodes from the given node.
DeleteNode
Removes the given node from the tree.

DeleteSelectedNodes
Removes all currently selected nodes from the tree.

Destroy
Destructor of the control.

DetermineHiddenChildrenFlag
Determines whether all children of a given node are hidden.

DetermineHiddenChildrenFlagAllNodes
Determines whether all children of all nodes are hidden.

DetermineHitPositionLTR
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineHitPositionRTL
Determines the hit position within a node with left-to-right and right-to-left orientation.

DetermineNextCheckState
Not documented.

DetermineScrollDirections
Not documented.

DoAdvancedHeaderDraw
Not documented.

DoAfterCellPaint
Not documented.

DoAfterItemErase
Not documented.

DoAfterItemPaint
Not documented.

DoAfterPaint
Not documented.

DoAutoScroll
Enables or disables the auto scroll timer.

DoBeforeCellPaint
Not documented.

DoBeforeDrag
Not documented.

- **DoBeforeItemErase**
  Not documented.

- **DoBeforeItemPaint**
  Not documented.

- **DoBeforePaint**
  Not documented.

- **DoCancelEdit**
  Called when the tree should stop editing without accepting changed values.

- **DoCanEdit**
  Not documented.

- **DoChange**
  Not documented.

- **DoCheckClick**
  Not documented.

- **DoChecked**
  Not documented.

- **DoChecking**
  Not documented.

- **DoCollapsed**
  Not documented.

- **DoCollapsing**
  Not documented.

- **DoColumnClick**
  Not documented.

- **DoColumnDbClick**
  Not documented.

- **DoColumnResize**
  Not documented.

- **DoCompare**
  Not documented.

- **DoCreateDataObject**
  Not documented.

- **DoCreateDragManager**
Not documented.

DoCreateEditor
Not documented.

DoDragDrop
Not documented.

DoDragExpand
Not documented.

DoDragging
Internal method which handles drag' drop.

DoDragOver
Not documented.

DoEdit
Initiates editing of the currently set focused column and edit node.

DoEndDrag
Not documented.

DoEndEdit
Stops the current edit operation and takes over the new content.

DoExpanded
Not documented.

DoExpanding
Not documented.

DoFocusChange
Not documented.

DoFocusChanging
Not documented.

DoFocusNode
Internal method to set the focused node.

DoFreeNode
Not documented.

DoGetAnimationType
Determines the type of animation to be used.

DoGetCursor
Not documented.

DoGetHeaderCursor
Not documented.
DoGetImageIndex
Not documented.
DoGetLineStyle
Not documented.
DoGetNodeHint
Not documented.
DoGetNodeTooltip
Not documented.
DoGetNodeWidth
Overridable method which always returns 0.
DoGetPopupMenu
Overridable method which triggers the OnGetPopup event.
DoGetUserClipboardFormats
Not documented.
DoHeaderClick
Not documented.
DoHeaderDoubleClick
Not documented.
DoHeaderDragged
Not documented.
DoHeaderDraggedOut
Not documented.
DoHeaderDragging
Not documented.
DoHeaderDraw
Not documented.
DoHeaderDrawQueryElements
Not documented.
DoHeaderMouseDown
Not documented.
DoHeaderMouseMove
Not documented.
DoHeaderMouseUp
Not documented.
DoHotChange
Not documented.

- DoIncrementalSearch
  Not documented.
- DoInitChildren
  Not documented.
- DoInitNode
  Not documented.
- DoKeyAction
  Not documented.
- DoLoadUserData
  Not documented.
- DoMeasureItem
  Not documented.
- DoNodeCopied
  Not documented.
- DoNodeCopying
  Not documented.
- DoNodeMoved
  Not documented.
- DoNodeMoving
  Not documented.
- DoPaintBackground
  Not documented.
- DoPaintDropMark
  Overridable method which draws the small line on top of a nodes image depending on the current drop state.
- DoPaintNode
  Overridable method which does nothing.
- DoPopupMenu
  Overridable method which shows the popup menu for the given node.
- DoRenderOLEDData
  Not documented.
- DoReset
  Not documented.
- DoSaveUserData
Not documented.

**DoScroll**
Overridable method which triggers the `OnScroll` event.

**DoSetOffsetXY**
Internal core routine to set the tree's scroll position.

**DoShowScrollbar**
Not documented.

**DoStartDrag**
Not documented.

**DoStateChange**
Not documented.

**DoStructureChange**
Not documented.

**DoTimerScroll**
Callback method which is triggered whenever the scroll timer fires.

**DoUpdating**
Not documented.

**DoValidateCache**
Not documented.

**DragCanceled**
Called by the VCL when a drag'n drop operation was canceled by the user.

**DragDrop**
Helper method, which is used when a drag operation is finished.

**DragEnter**
Not documented.

**DragFinished**
Called when a drag operation is finished (accepted or cancelled).

**Dragging**
Returns true if a drag'n drop operation is in progress.

**DragLeave**
Not documented.

**DragOver**
Not documented.

**DrawDottedHLine**
Not documented.

- **DrawDottedVLine**
  Not documented.

- **EditNode**
  Starts editing the given node if allowed to.

- **EndEditNode**
  Stops node editing if it was started before.

- **EndSynch**
  Counterpart to **BeginSynch**.

- **EndUpdate**
  Resets the update lock set by **BeginUpdate**.

- **ExecuteAction**
  Not documented.

- **FindNodeInSelection**
  Helper method to find the given node in the current selection.

- **FinishChunkHeader**
  Not documented.

- **FinishCutOrCopy**
  Stops any pending cut or copy clipboard operation.

- **FlushClipboard**
  Renders all pending clipboard data.

- **FontChanged**
  Not documented.

- **FullCollapse**
  Collapses all nodes in the tree.

- **FullExpand**
  Expands all nodes in the tree.

- **GetBorderDimensions**
  Not documented.

- **GetCheckImage**
  Not documented.

- **GetCheckImageListFor**
  Not documented.

- **GetColumnClass**
  Returns the class to be used to manage columns in the tree.
GetControlsAlignment
   Not documented.

GetDisplayRect
   Returns the visible region used by the given node in client coordinates.

GetFirst
   Group of node navigation functions.

GetFirstChild
   Group of node navigation functions.

GetFirstCutCopy
   Group of node navigation functions.

GetFirstInitialized
   Group of node navigation functions.

GetFirstNoInit
   Group of node navigation functions.

GetFirstSelected
   Group of node navigation functions.

GetFirstVisible
   Group of node navigation functions.

GetFirstVisibleChild
   Group of node navigation functions.

GetFirstVisibleChildNoInit
   Group of node navigation functions.

GetFirstVisibleNoInit
   Group of node navigation functions.

GetHeaderClass
   Returns the header class to be used by the tree.

GetHintWindowClass
   Not documented.

GetHitTestInfoAt
   Returns information about the node at the given position.

GetImageIndex
   Not documented.
GetLast
Group of node navigation functions.

GetFirstChild
Group of node navigation functions.

GetFirstChildNoInit
Group of node navigation functions.

GetFirstChildInitialized
Group of node navigation functions.

GetFirstChildNoInit
Group of node navigation functions.

GetFirstChildVisible
Group of node navigation functions.

GetFirstChildVisibleChild
Group of node navigation functions.

GetFirstChildVisibleChildNoInit
Group of node navigation functions.

GetFirstChildVisibleNoInit
Group of node navigation functions.

GetMaxColumnWidth
  Returns the width of the largest node in the given column.

GetMaxRightExtend
  Determines the maximum width of the currently visible part of the tree.

GetNativeClipboardFormats
  Used to let descendants and the application add their own supported clipboard formats.

GetNext
Group of node navigation functions.

GetNextChecked
  Not documented.

GetNextCutCopy
Group of node navigation functions.

GetNextInitialized
Group of node navigation functions.

GetNextNoInit
Group of node navigation functions.
- **GetNextSelected**
  Group of node navigation functions.
- **GetNextSibling**
  Group of node navigation functions.
- **GetNextVisible**
  Group of node navigation functions.
- **GetNextVisibleNoInit**
  Group of node navigation functions.
- **GetNextVisibleSibling**
  Group of node navigation functions.
- **GetNextVisibleSiblingNoInit**
  Group of node navigation functions.
- **GetNodeAt**
  Not documented.
- **GetNodeData**
  Returns the address of the user data area of the given node.
- **GetNodeLevel**
  Returns the indentation level of the given node.
- **GetOptionsClass**
  Customization helper to determine which options class the tree should use.
- **GetPrevious**
  Group of node navigation functions.
- **GetPreviousInitialized**
  Group of node navigation functions.
- **GetPreviousNoInit**
  Group of node navigation functions.
- **GetPreviousSibling**
  Group of node navigation functions.
- **GetPreviousVisible**
  Group of node navigation functions.
- **GetPreviousVisibleNoInit**
  Group of node navigation functions.
- **GetPreviousVisibleSibling**
  Group of node navigation functions.
- **GetPreviousVisibleSiblingNoInit**
  Group of node navigation functions.

- **GetSortedCutCopySet**
  Returns a sorted list of nodes, which are marked for cut or copy clipboard operation.

- **GetSortedSelection**
  Returns a sorted list of all currently selected nodes.

- **GetTextInfo**
  Helper method for node editors, hints etc.

- **GetTreeFromDataObject**
  OLE drag’n drop and clipboard support method.

- **GetTreeRect**
  Returns the size of the virtual tree image.

- **GetVisibleParent**
  Returns the first (nearest) parent node, which is visible.

- **HandleHotTrack**
  Not documented.

- **HandleIncrementalSearch**
  Not documented.

- **HandleMouseDbClick**
  Not documented.

- **HandleMouseDown**
  Not documented.

- **HandleMouseUp**
  Not documented.

- **HasAsParent**
  Determines if the given node has got another node as one of its parents.

- **HasImage**
  Not documented.

- **HasPopupMenu**
  Determines whether there is a pop up menu assigned to the tree.

- **InitChildren**
  Not documented.

- **InitNode**
Not documented.

- **InsertNode**
  Inserts a new node and returns it to the caller.

- **InternalAddFromStream**
  Not documented.

- **InternalAddToSelection**
  Not documented.

- **InternalCacheNode**
  Not documented.

- **InternalClearSelection**
  Not documented.

- **InternalConnectNode**
  Not documented.

- **InternalData**
  Returns the address of the internal data for a tree class.

- **InternalDisconnectNode**
  Not documented.

- **InternalRemoveFromSelection**
  Not documented.

- **InvalidateCache**
  Empties the internal node cache and marks it as invalid.

- **InvalidateChildren**
  Invalidates all children of the given node.

- **InvalidateColumn**
  Invalidates the client area part of a column.

- **InvalidateNode**
  Invalidates the given node.

- **InvalidateToBottom**
  Invalidates the client area starting with the top position of the given node.

- **InvertSelection**
  Inverts the current selection.

- **IsEditing**
  Tells the caller whether the tree is currently in edit mode.

- **IsMouseSelecting**
Tell the caller whether the tree is currently in draw selection mode.

- **IterateSubtree**
  - Iterator method to go through all nodes of a given sub tree.

- **Loaded**
  - Not documented.

- **LoadFromFile**
  - Loads previously streamed out tree data back in again.

- **LoadFromStream**
  - Loads previously streamed out tree data back in again.

- **MainColumnChanged**
  - Not documented.

- **MarkCutCopyNodes**
  - Not documented.

- **MeasureItemHeight**
  - Not documented.

- **MouseMove**
  - Not documented.

- **MoveTo**
  - Moves **Source** and all its child nodes to **Target**.

- **Notification**
  - Not documented.

- **OriginalWMNCPaint**
  - Not documented.

- **Paint**
  - TControl's Paint method used here to display the tree.

- **PaintCheckImage**
  - Not documented.

- **PaintImage**
  - Not documented.

- **PaintNodeButton**
  - Not documented.

- **PaintSelectionRectangle**
  - Not documented.

- **PaintTree**
  - Main paint routine for the tree image.
PaintTreeLines
Not documented.
PanningWindowProc
Not documented.
PasteFromClipboard
Inserts the content of the clipboard into the tree.
PrepareDragImage
Not documented.
Print
Not documented.
ProcessDrop
Helper method to ease OLE drag'n drop operations.
ProcessOLEData
Takes serialized OLE tree data and reconstructs the former structure.
ReadChunk
Not documented.
ReadNode
Not documented.
RedirectFontChangeEvent
Not documented.
ReinitChildren
Forces all child nodes of Node to be reinitialized.
ReinitNode
Forces a reinitialization of the given node.
RemoveFromSelection
Removes the given node from the current selection.
RenderOLEData
Renders pending OLE data.
RepaintNode
Causes the treeview to repaint the given node.
ResetNode
Resets the given node to uninitialized.
ResetRangeAnchor
Not documented.
RestoreFontChangeEvent
Not documented.

- **SaveToFile**
  Saves the entire content of the tree into a file or stream.

- **SaveToStream**
  Saves the entire content of the tree into a file or stream.

- **ScrollIntoView**
  Scrolls the tree so that the given node comes in the client area.

- **SelectAll**
  Selects all nodes in the tree.

- **SelectNodes**
  Selects a range of nodes.

- **SetBiDiMode**
  Not documented.

- **SetFocusedNodeAndColumn**
  Not documented.

- **SkipNode**
  Not documented.

- **Sort**
  Sorts the given node.

- **SortTree**
  Sorts the entire tree view.

- **StartWheelPanning**
  Not documented.

- **StopWheelPanning**
  Not documented.

- **StructureChange**
  Not documented.

- **SuggestDropEffect**
  Not documented.

- **ToggleNode**
  Changes a node's expand state to the opposite state.

- **ToggleSelection**
  Toggles the selection state of a range of nodes.

- **UnselectAll**
  Deselects a range of nodes.
UpdateAction
Not documented.
UpdateDesigner
Not documented.
UpdateEditBounds
Not documented.
UpdateHeaderRect
Not documented.
UpdateHorizontalScrollBar
Applies changes to the horizontal and vertical scrollbars.
UpdateScrollBars
Applies changes to the horizontal and vertical scrollbars.
UpdateVerticalScrollBar
Applies changes to the horizontal and vertical scrollbars.
UpdateWindowAndDragImage
Not documented.
UseRightToLeftReading
Helper method for right-to-left layout.
ValidateCache
Initiates the validation of the internal node cache.
ValidateChildren
Validates all children of a given node.
ValidateNode
Validates a given node.
ValidateNodeDataSize
Helper method for node data size initialization.
WndProc
Redirected window procedure to do some special processing.
WriteChunks
Writes the core chunks for the given node to the given stream.
WriteNode
Writes the cover (envelop) chunk for the given node to the given stream.
TVirtualDrawTree Class Properties

Properties

- **Action**
  Not documented.

- **Align**
  Not documented.

- **Alignment**
  Determines the horizontal alignment of text if no columns are defined.

- **Anchors**
  Not documented.

- **AnimationDuration**
  Determines the maximum duration the tree can use to play an animation.

- **AutoExpandDelay**
  Time delay after which a node gets expanded if it is the current drop target.

- **AutoScrollDelay**
  Time which determines when auto scrolling should start.

- **AutoScrollInterval**
  Time interval between scroll events when doing auto scroll.

- **Background**
  Holds a background image for the tree.

- **BackgroundOffsetX**
  Horizontal offset of the background image.

- **BackgroundOffsetY**
  Vertical offset of the background image.

- **BevelEdges**
  Not documented.

- **BevelInner**
Not documented.

- **BevelKind**  
  Not documented.

- **BevelOuter**  
  Not documented.

- **BevelWidth**  
  Not documented.

- **BiDiMode**  
  Not documented.

- **BorderStyle**  
  Same as TForm.BorderStyle.

- **BorderWidth**  
  Not documented.

- **ButtonFillMode**  
  Determines how to fill the background of the node buttons.

- **ButtonStyle**  
  Determines the look of node buttons.

- **Canvas**  
  Not documented.

- **ChangeDelay**  
  Time which determines when the OnChange event should be triggered after the actual change event.

- **CheckImageKind**  
  Determines which images should be used for checkboxes and radio buttons.

- **ClipboardFormats**  
  Special class to keep a list of clipboard format descriptions.

- **Color**  
  Not documented.

- **Colors**  
  A collection of colors used in the tree.

- **Constraints**  
  Not documented.

- **Ctl3D**  
  Not documented.
CustomCheckImages
Assign your own image list to get the check images you like most.

DefaultNodeHeight
Read or set the height new nodes get as initial value.

DefaultPasteMode
Read or set the value, which determines where to add pasted nodes to.

DragCursor
Not documented.

DragHeight
Read or set the vertical limit of the internal drag image.

DragImageKind
Read or set what should be shown in the drag image.

DragKind
Not documented.

DragMode
Not documented.

DragOperations
Read or set which drag operations may be allowed in the tree.

DragType
Read or set which subsystem should be used for dragging.

DragWidth
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode
Read or set how multiselection with the mouse is to be visualized.

EditDelay
Read or set the maximum time between two single clicks on the same node, which should start node editing.

Enabled
Not documented.

Font
Same as TWinControl.Font.

Header
Provides access to the header instance.

HintAnimation
Read or set the current hint animation type.

- **HintMode**
  Read or set what type of hint you want for the tree view.

- **HotCursor**
  Read or set which cursor should be used for hot nodes.

- **Images**
  Read or set the tree's normal image list.

- **IncrementalSearch**
  Read or set the current incremental search mode.

- **IncrementalSearchDirection**
  Read or set the direction to be used for incremental search.

- **IncrementalSearchStart**
  Read or set where to start incremental search.

- **IncrementalSearchTimeout**
  Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

- **Indent**
  Read or set the indentation amount for node levels.

- **LineMode**
  Read or set the mode of the tree lines.

- **LineStyle**
  Read or set the mode of the tree lines.

- **Margin**
  Read or set the tree's node margin.

- **NodeAlignment**
  Read or set the node alignment value.

- **NodeDataSize**
  Read or set the extra data size for each node.

- **OnClick**
  Not documented.

- **OnDbClick**
  Not documented.

- **OnEndDock**
  Not documented.

- **OnEndDrag**
Not documented.

- OnEnter
Not documented.

- OnExit
Not documented.

- OnKeyDown
Not documented.

- OnKeyPress
Not documented.

- OnKeyUp
Not documented.

- OnMouseDown
Not documented.

- OnMouseMove
Not documented.

- OnMouseUp
Not documented.

- OnMouseWheel
Not documented.

- OnResize
Not documented.

- OnStartDock
Not documented.

- ParentBiDiMode
Not documented.

- ParentColor
Not documented.

- ParentCtl3D
Not documented.

- ParentFont
Not documented.

- ParentShowHint
Not documented.

- PopupMenu
Not documented.
RootNodeCount
Read or set the number of nodes on the top level.

ScrollBarOptions
Reference to the scroll bar options class.

SelectionBlendFactor
Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

SelectionCurveRadius
Read or set the current corner radius for node selection rectangles.

ShowHint
Not documented.

StateImages
Reference to the images list which is used for the state images.

TabOrder
Not documented.

TabStop
Not documented.

TextMargin
Read or set the distance of the node caption to its borders.

TreeOptions
Reference to the tree's options.

Visible
Not documented.

WantTabs
Read or set whether the tree wants to process tabs on its own.

TBaseVirtualTree Class

Alignment
Determines the horizontal alignment of text if no columns are defined.

AnimationDuration
Determines the maximum duration the tree can use to play an animation.

AutoExpandDelay
Time delay after which a node gets expanded if it is the current drop target.
AutoScrollDelay
Time which determines when auto scrolling should start.

AutoScrollInterval
Time interval between scroll events when doing auto scroll.

Background
Holds a background image for the tree.

BackgroundOffsetX
Horizontal offset of the background image.

BackgroundOffsetY
Vertical offset of the background image.

BorderStyle
Same as TForm.BorderStyle.

ButtonFillMode
Determines how to fill the background of the node buttons.

ButtonStyle
Determines the look of node buttons.

ChangeDelay
Time which determines when theOnChange event should be triggered after the actual change event.

CheckImageKind
Determines which images should be used for checkboxes and radio buttons.

CheckImages
Not documented.

CheckState
Read or set the check state of a node.

CheckType
Read or set the check type of a node.

ChildCount
Read or set the number of child nodes of a node.

ChildrenInitialized
Read whether a node's child count has been initialized already.

ClipboardFormats
Special class to keep a list of clipboard format descriptions.

Colors
A collection of colors used in the tree.

- **CustomCheckImages**
  Assign your own image list to get the check images you like most.

- **DefaultNodeHeight**
  Read or set the height new nodes get as initial value.

- **DefaultPasteMode**
  Read or set the value, which determines where to add pasted nodes to.

- **DragHeight**
  Read or set the vertical limit of the internal drag image.

- **DragImage**
  Holds the instance of the internal drag image.

- **DragImageKind**
  Read or set what should be shown in the drag image.

- **DragManager**
  Holds the reference to the internal drag manager.

- **DragOperations**
  Read or set which drag operations may be allowed in the tree.

- **DragSelection**
  Keeps a temporary list of nodes during drag'n drop.

- **DragType**
  Read or set which subsystem should be used for dragging.

- **DragWidth**
  Read or set the horizontal limit of the internal drag image.

- **DrawSelectionMode**
  Read or set how multiselection with the mouse is to be visualized.

- **DropTargetNode**
  Contains the current drop target node if the tree is currently the target of a drag'n drop operation.

- **EditColumn**
  Not documented.

- **EditDelay**
  Read or set the maximum time between two single clicks on the same node, which should start node editing.

- **EditLink**
Keeps a reference to the internal edit link during a node edit operation.

- **Expanded**
  Read or set the expanded state of a particular node.

- **FocusedColumn**
  Read or set the currently focused column.

- **FocusedNode**
  Read or set the currently focused node.

- **Font**
  Same as TWinControl.Font.

- **FullyVisible**
  Read or set whether a node is fully visible or not.

- **HasChildren**
  Read or set whether a node has got children.

- **Header**
  Provides access to the header instance.

- **HeaderRect**
  Returns the non-client-area rectangle used for the header.

- **HintAnimation**
  Read or set the current hint animation type.

- **HintMode**
  Read or set what type of hint you want for the tree view.

- **HotCursor**
  Read or set which cursor should be used for hot nodes.

- **HotNode**
  Read, which node is currently the hot node.

- **Images**
  Read or set the tree's normal image list.

- **IncrementalSearch**
  Read or set the current incremental search mode.

- **IncrementalSearchDirection**
  Read or set the direction to be used for incremental search.

- **IncrementalSearchStart**
  Read or set where to start incremental search.

- **IncrementalSearchTimeout**
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

- **Indent**
  Read or set the indentation amount for node levels.

- **IsDisabled**
  Read or set the enabled state of the given node.

- **IsVisible**
  Read or set the visibility state of the given node.

- **LastClickPos**
  Used for retained drag start and wheel mouse scrolling.

- **LastDropMode**
  Read how the last drop operation finished.

- **LineMode**
  Read or set the mode of the tree lines.

- **LineStyle**
  Read or set the mode of the tree lines.

- **Margin**
  Read or set the tree's node margin.

- **MultiLine**
  Read or toggle the multiline feature for a given node.

- **NodeAlignment**
  Read or set the node alignment value.

- **NodeDataSize**
  Read or set the extra data size for each node.

- **NodeHeight**
  Read or set a node's height.

- **NodeParent**
  Read or set a node's parent node.

- **OffsetX**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetXY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **RootNode**
Reference to the internal root node which is the anchor of the entire tree node hierarchy.

- **RootNodeCount**
  Read or set the number of nodes on the top level.

- **ScrollBarOptions**
  Reference to the scroll bar options class.

- **SearchBuffer**
  Current input string for incremental search.

- **Selected**
  Property to modify or determine the selection state of a node.

- **SelectedCount**
  Contains the number of selected nodes.

- **SelectionBlendFactor**
  Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

- **SelectionCurveRadius**
  Read or set the current corner radius for node selection rectangles.

- **StateImages**
  Reference to the images list which is used for the state images.

- **TextMargin**
  Read or set the distance of the node caption to its borders.

- **TopNode**
  The top node is the node which is currently at the top border of the client area.

- **TotalCount**
  Returns the number of nodes in the tree.

- **TotalInternalDataSize**
  Keeps the currently accumulated data size for one node.

- **TreeOptions**
  Reference to the tree's options.

- **TreeStates**
  Property which keeps a set of flags which indicate current operation and states of the tree.

- **UpdateCount**
  Not documented.
- **VerticalAlignment**
  Used to set a node's vertical button alignment with regard to the entire node rectangle.

- **VisibleCount**
  Number of currently visible nodes.

- **VisiblePath**
  Property to set or determine a node parent's expand states.

- **WantTabs**
  Read or set whether the tree wants to process tabs on its own.
See Also
TVirtualDrawTree.CustomCheckImages Property

See Also
TCheckImageKind
See Also
TVirtualDrawTree.DefaultPasteMode Property

See Also
TVTNodeAttachMode
See Also
TVirtualDrawTree.EditDelay Property

See Also
Editors and editing
See Also
TVirtualDrawTree.Header Property

See Also
TVTHeader
See Also
TVirtualDrawTree.HotCursor Property

See Also
HotNode, TVTPaintOptions
See Also
TVirtualDrawTree.Images Property

See Also
StateImages, CheckImages
See Also

TVirtualDrawTree.IncrementalSearch Property

See Also

IncrementalSearchDirection, IncrementalSearchStart, IncrementalSearchTimeout
See Also
TVirtualDrawTree.IncrementalSearchDirection Property

See Also
IncrementalSearch, IncrementalSearchStart, IncrementalSearchTime123out
See Also
TVirtualDrawTree.IncrementalSearchStart Property

See Also
IncrementalSearch, IncrementalSearchDirection, IncrementalSearchTimeout
See Also
TVirtualDrawTree.IncrementalSearchTimeout Property

See Also
IncrementalSearch, IncrementalSearchDirection, IncrementalSearchStart
See Also

TVirtualDrawTree.Margin Property

See Also

TVirtualStringTree.TextMargin
See Also
TVirtualDrawTree.NodeAlignment Property

See Also
TVirtualNode
See Also
TVirtualDrawTree.NodeDataSize Property

See Also
Data handling
See Also
TVirtualDrawTree.OnAdvancedHeaderDraw Event

See Also
OnHeaderDrawQueryElements, OnHeaderDraw
See Also

TVirtualDrawTree.OnAfterCellPaint Event

See Also

Paint cycles and stages
See Also
TVirtualDrawTree.OnAfterItemErase Event

See Also
Paint cycles and stages
See Also
TVirtualDrawTree.OnAfterItemPaint Event

See Also
Paint cycles and stages
See Also
TVirtualDrawTree.OnAfterPaint Event

See Also
Paint cycles and stages
See Also
TVirtualDrawTree.OnBeforeCellPaint Event

See Also
Paint cycles and stages
See Also
TVirtualDrawTree.OnBeforeItemErase Event

See Also
Paint cycles and stages
See Also
TVirtualDrawTree.OnBeforeItemPaint Event

See Also
Paint cycles and stages
See Also

TVirtualDrawTree.OnBeforePaint Event

See Also

Paint cycles and stages
See Also
TVirtualDrawTree.OnColumnClick Event

See Also
OnHeaderClick
See Also
TVirtualDrawTree.OnColumnDblClick Event

See Also
OnColumnClick, OnHeaderDblClick
See Also
TVirtualDrawTree.OnCompareNodes Event

See Also
SortTree, Sort
See Also
TVirtualDrawTree.OnCreateEditor Event

See Also
Editors and editing
See Also
TVirtualDrawTree.OnDragOver Event

See Also
OnDragDrop
See Also
TVirtualDrawTree.OnEdited Event

See Also
Editors and editing
See Also

TVirtualDrawTree.OnEditing Event

See Also

Editors and editing
See Also
TVirtualDrawTree.OnGetLineStyle Event

See Also
PrepareBitmaps
See Also
TVirtualDrawTree.OnGetNodeDataSize Event

See Also
NodeDataSize, Data handling
See Also
TVirtualDrawTree.OnHeaderClick Event

See Also
SortColumn, SortDirection
See Also

TVirtualDrawTree. OnHeaderDblClick Event

See Also

OnHeaderClick
See Also
TVirtualDrawTree.OnHeaderDrawQueryElements Event

See Also
OnAdvancedHeaderDraw
See Also
TVirtualDrawTree.OnInitChildren Event

See Also
The virtual paradigm
See Also
TVirtualDrawTree.OnInitNode Event

See Also
The virtual paradigm
See Also

TVirtualDrawTree.OnLoadNode Event

See Also

OnSaveNode, LoadFromStream, SaveToStream, AddFromStream, VTTreeStreamVersion, TVTHeader.LoadFromStream, TVTHeader.SaveToStream
See Also
TVirtualDrawTree.OnMeasureItem Event

See Also
InvalidateNode, vsHeightMeasured
See Also

TVirtualDrawTree.OnResetNode Event

See Also

ResetNode
See Also
TVirtualDrawTree.OnSaveNode Event

See Also
OnLoadNode, LoadFromStream, SaveToStream, AddFromStream, VTTreeStreamVersion, TVTHeader.LoadFromStream, TVTHeader.SaveToStream
See Also
TVirtualDrawTree.OnScroll Event

See Also
OffsetXY
See Also
TVirtualDrawTree.SelectionBlendFactor Property

See Also
DrawSelectionMode, TVTPaintOptions
See Also
TVirtualDrawTree.SelectionCurveRadius Property

See Also
SelectionBlendFactor, DrawSelectionMode, TVTPaintOptions
See Also

TVirtualDrawTree.StateImages Property

See Also

CheckImages, Images
See Also
TVirtualDrawTree.TextMargin Property

See Also
Margin
TVirtualStringTree Class Events

Events

- **OnAdvancedHeaderDraw**
  Header paint support event.

- **OnAfterCellPaint**
  Paint support event.

- **OnAfterItemErase**
  Paint support event.

- **OnAfterItemPaint**
  Paint support event.

- **OnAfterPaint**
  Paint support event.

- **OnBeforeCellPaint**
  Paint support event.

- **OnBeforeItemErase**
  Paint support event.

- **OnBeforeItemPaint**
  Paint support event.

- **OnBeforePaint**
  Paint support event.

- **OnChange**
  Navigation support event.

- **OnChecked**
  Check support event.

- **OnChecking**
  Check support event.

- **OnCollapsed**
  Miscellaneous event.
- **OnCollapsing**
  Miscellaneous event.
- **OnColumnClick**
  Header and column support event.
- **OnColumnDbIClick**
  Header and column support event.
- **OnColumnResize**
  Header and column support routine.
- **OnCompareNodes**
  Sort and search support event.
- **OnCreateDataObject**
  Drag'n drop support event.
- **OnCreateDragManager**
  Drag'n drop support event.
- **OnCreateEditor**
  Editing support event.
- **OnDragAllowed**
  Drag'n drop support event.
- **OnDragDrop**
  Drag'n drop support event.
- **OnDragOver**
  Drag'n drop support event.
- **OnEditCancelled**
  Editing support event.
- **OnEdited**
  Editing support event.
- **OnEditing**
  Editing support event.
- **OnExpanded**
  Miscellaneous event.
- **OnExpanding**
  Miscellaneous event.
- **OnFocusChanged**
  Navigation support event.
- **OnFocusChanging**
Navigation support event.

- **OnFreeNode**
  Data management node.

- **OnGetCellIsEmpty**
  Triggered when the tree control needs to know whether a given column is empty.

- **OnGetCursor**
  Miscellaneous event.

- **OnGetHeaderCursor**
  Header and column support event.

- **OnGetHelpContext**
  Miscellaneous event.

- **OnGetHint**
  Virtual string tree event to query for a custom hint text.

- **OnGetImageIndex**
  Display management event.

- **OnGetImageIndexEx**
  Not documented.

- **OnGetLineStyle**
  Display management event.

- **OnGetNodeDataSize**
  Data management event.

- **OnGetPopupMenu**
  Miscellaneous event.

- **OnGetText**
  Virtual string tree event to query for a node's normal or static text.

- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.

- **OnHeaderClick**
  Header & column support event.

- **OnHeaderDb1Click**
  Header & column support event.

- **OnHeaderDragged**
  Header & column support event.

- **OnHeaderDraggedOut**
Header & column support event.

OnHeaderDragging
Header & column support event.

OnHeaderDraw
Header & column support event.

OnHeaderDrawQueryElements
Header & column support event.

OnHeaderMouseDown
Header & column support event.

OnHeaderMouseMove
Header & column support event.

OnHeaderMouseUp
Header & column support event.

OnHotChange
Navigation support event.

OnIncrementalSearch
Miscellaneous event.

OnInitChildren
Node management event.

OnInitNode
Node management event.

OnKeyAction
Miscellaneous event.

OnLoadNode
Streaming support event.

OnMeasureItem
Miscellaneous event.

OnNewText
Virtual string tree event to pass edited text.

OnNodeCopied
Miscellaneous event.

OnNodeCopying
Miscellaneous event.

OnNodeMoved
Miscellaneous event.
TCustomVirtualStringTree Class

- **OnGetHint**
  Virtual string tree event to query for a custom hint text.

- **OnGetText**
  Virtual string tree event to query for a node's normal or static text.

- **OnNewText**
  Virtual string tree event to pass edited text.

- **OnPaintText**
  Event to change text formatting for particular nodes.
OnShortenString
String tree event for custom handling of string abbreviations.

TBaseVirtualTree Class

OnAdvancedHeaderDraw
Header paint support event.
OnAfterCellPaint
Paint support event.
OnAfterItemErase
Paint support event.
OnAfterItemPaint
Paint support event.
OnAfterPaint
Paint support event.
OnBeforeCellPaint
Paint support event.
OnBeforeItemErase
Paint support event.
OnBeforeItemPaint
Paint support event.
OnBeforePaint
Paint support event.
OnChange
Navigation support event.
OnChecked
Check support event.
OnChecking
Check support event.
OnCollapsed
Miscellaneous event.
OnCollapsing
Miscellaneous event.
OnColumnClick
Header and column support event.
OnColumnDb1Click
Header and column support event.

OnColumnResize
Header and column support routine.

OnCompareNodes
Sort and search support event.

OnCreateDataObject
Drag'n drop support event.

OnCreateDragManager
Drag'n drop support event.

OnCreateEditor
Editing support event.

OnDragAllowed
Drag'n drop support event.

OnDragDrop
Drag'n drop support event.

OnDragOver
Drag'n drop support event.

OnEditCancelled
Editing support event.

OnEdited
Editing support event.

OnEditing
Editing support event.

OnExpanded
Miscellaneous event.

OnExpanding
Miscellaneous event.

OnFocusChanged
Navigation support event.

OnFocusChanging
Navigation support event.

OnFreeNode
Data management node.

OnGetCellIsEmplty
Triggered when the tree control needs to know whether a given column is empty.

- **OnGetCursor**
  Miscellaneous event.

- **OnGetHeaderCursor**
  Header and column support event.

- **OnGetHelpContext**
  Miscellaneous event.

- **OnGetImageIndex**
  Display management event.

- **OnGetImageIndexEx**
  Not documented.

- **OnGetLineStyle**
  Display management event.

- **OnGetNodeDataSize**
  Data management event.

- **OnGetPopupMenu**
  Miscellaneous event.

- **OnGetUserClipboardFormats**
  Drag'n drop and clipboard support event.

- **OnHeaderClick**
  Header & column support event.

- **OnHeaderDb1Click**
  Header & column support event.

- **OnHeaderDragged**
  Header & column support event.

- **OnHeaderDraggedOut**
  Header & column support event.

- **OnHeaderDragging**
  Header & column support event.

- **OnHeaderDraw**
  Header & column support event.

- **OnHeaderDrawQueryElements**
  Header & column support event.

- **OnHeaderMouseDown**
Header & column support event.

- **OnHeaderMouseMove**
- **OnHeaderMouseUp**
- **OnHotChange**
  - Navigation support event.
- **OnIncrementalSearch**
  - Miscellaneous event.
- **OnInitChildren**
  - Node management event.
- **OnInitNode**
  - Node management event.
- **OnKeyAction**
  - Miscellaneous event.
- **OnLoadNode**
  - Streaming support event.
- **OnMeasureItem**
  - Miscellaneous event.
- **OnNodeCopied**
  - Miscellaneous event.
- **OnNodeCopying**
  - Miscellaneous event.
- **OnNodeMoved**
  - Miscellaneous event.
- **OnNodeMoving**
  - Miscellaneous event.
- **OnPaintBackground**
  - Paint support event.
- **OnRenderOLEData**
  - Drag'n drop and clipboard support event.
- **OnResetNode**
  - Node management event.
- **OnSaveNode**
  - Streaming support event.
OnScroll
Miscellaneous event.

OnShowScrollbar
Not documented.

OnStateChange
Miscellaneous event.

OnStructureChange
Miscellaneous event.

OnUpdating
Miscellaneous event.
TVirtualStringTree Class Methods

**Methods**

- `GetOptionsClass`
  - Customization helper to determine which options class the tree should use.

TCustomVirtualStringTree Class

- `AdjustPaintCellRect`
  - Method which can be used by descendants to adjust the given rectangle during a paint cycle.
- `CalculateTextWidth`
  - Not documented.
- `ColumnIsEmpty`
  - Used to determine if a cell is considered as being empty.
- `ComputeNodeHeight`
  - Not documented.
- `ContentToClipboard`
  - Not documented.
- `ContentToHTML`
  - Not documented.
- `ContentToRTF`
  - Not documented.
- `ContentToText`
  - Not documented.
- `ContentToUnicode`
  - Not documented.
- `Create`
  - Constructor of the control
DefineProperties
Helper method to customize loading and saving persistent tree data.

DoCreateEditor
Not documented.

DoGetNodeHint
Not documented.

DoGetNodeTooltip
Not documented.

DoGetNodeWidth
Overridable method which always returns 0.

DoGetText
Not documented.

DoIncrementalSearch
Not documented.

DoNewText
Not documented.

DoPaintNode
Overridable method which does nothing.

DoPaintText
Not documented.

DoShortenString
Not documented.

DoTextDrawing
Not documented.

DoTextMeasuring
Not documented.

GetOptionsClass
Customization helper to determine which options class the tree should use.

GetTextInfo
Helper method for node editors, hints etc.

InternalData
Returns the address of the internal data for a tree class.

InvalidateNode
Invalidates the given node.
MainColumnChanged
Not documented.

Path
Not documented.

ReadChunk
Not documented.

ReadOldStringOptions
Not documented.

ReinitNode
Forces a reinitialization of the given node.

RenderOLEData
Renders pending OLE data.

WriteChunks
Writes the core chunks for the given node to the given stream.

TBaseVirtualTree Class

AbsoluteIndex
Reads the overall index of a node.

AddChild
Creates and adds a new child node to given node.

AddFromStream
Adds the content from the given stream to the given node.

AddToSelection
Adds one or more nodes to the current selection.

AdjustPaintCellRect
Used in descendants to modify the clip rectangle of the current column while painting a certain node.

AdjustPanningCursor
Loads the proper cursor which indicates into which direction scrolling is done.

AdviseChangeEvent
Used to register a delayed change event.

AllocateInternalDataArea
Registration method to allocate tree internal data per node.
Animate
Support method for animated actions in the tree view.

Assign
Used to copy properties from another Virtual Treeview.

BeginDrag
Starts an OLE drag'n drop operation.

BeginSynch
Enters the tree into a special synchronized mode.

BeginUpdate
Locks the tree view to perform several update operations.

CalculateSelectionRect
Support method for draw selection.

CanAutoScroll
Determines whether the tree can currently auto scroll its window.

CancelCutOrCopy
Cancels any pending cut or copy clipboard operation.

CancelEditNode
Cancel the current edit operation, if there is any.

CanEdit
Determines whether a node can be edited or not.

CanFocus
Support method to determine whether the tree window can receive the input focus.

CanShowDragImage
Determines whether a drag image should be shown.

Change
Central method called when a node's selection state changes.

ChangeScale
Helper method called by the VCL when control resizing is due.

CheckParentCheckState
Helper method for recursive check state changes.

Clear
Clears the tree and removes all nodes.

ClearChecked
Not documented.
ClearSelection
Removes all nodes from the current selection.

ClearTempCache
Helper method to clear the internal temporary node cache.

ColumnIsEmpty
Used to determine if a cell is considered as being empty.

CopyTo
Copies Source and all its child nodes to Target.

CopyToClipBoard
Copies all currently selected nodes to the clipboard.

CountLevelDifference
Determines the level difference of two nodes.

CountVisibleChildren
Determines the number of visible child nodes of the given node.

Create
Constructor of the control

CreateParams
Prepares the creation of the controls window handle.

CreateWnd
Initializes data, which depends on the window handle.

CutToClipBoard
Copies the currently selected nodes to the clipboard and removes them once a consumer has taken the data.

DefineProperties
Helper method to customize loading and saving persistent tree data.

DeleteChildren
Removes all child nodes from the given node.

DeleteNode
Removes the given node from the tree.

DeleteSelectedNodes
Removes all currently selected nodes form the tree.

Destroy
Destructor of the control.

DetermineHiddenChildrenFlag
Determines whether all children of a given node are hidden.
- **DetermineHiddenChildrenFlagAllNodes**: Determines whether all children of all nodes are hidden.
- **DetermineHitPositionLTR**: Determines the hit position within a node with left-to-right and right-to-left orientation.
- **DetermineHitPositionRTL**: Determines the hit position within a node with left-to-right and right-to-left orientation.
- **DetermineNextCheckState**: Not documented.
- **DetermineScrollDirections**: Not documented.
- **DoAdvancedHeaderDraw**: Not documented.
- **DoAfterCellPaint**: Not documented.
- **DoAfterItemErase**: Not documented.
- **DoAfterItemPaint**: Not documented.
- **DoAfterPaint**: Not documented.
- **DoAutoScroll**: Enables or disables the auto scroll timer.
- **DoBeforeCellPaint**: Not documented.
- **DoBeforeDrag**: Not documented.
- **DoBeforeItemErase**: Not documented.
- **DoBeforeItemPaint**: Not documented.
- **DoBeforePaint**: Not documented.
- **DoCancelEdit**
Called when the tree should stop editing without accepting changed values.

- DoCanEdit
  Not documented.
- DoChange
  Not documented.
- DoCheckClick
  Not documented.
- DoChecked
  Not documented.
- DoChecking
  Not documented.
- DoCollapsed
  Not documented.
- DoCollapsing
  Not documented.
- DoColumnClick
  Not documented.
- DoColumnDbIClick
  Not documented.
- DoColumnResize
  Not documented.
- DoCompare
  Not documented.
- DoCreateDataObject
  Not documented.
- DoCreateDragManager
  Not documented.
- DoCreateEditor
  Not documented.
- DoDragDrop
  Not documented.
- DoDragExpand
  Not documented.
- DoDragging
Internal method which handles drag' drop.

- DoDragOver
  Not documented.
- DoEdit
  Initiates editing of the currently set focused column and edit node.
- DoEndDrag
  Not documented.
- DoEndEdit
  Stops the current edit operation and takes over the new content.
- DoExpanded
  Not documented.
- DoExpanding
  Not documented.
- DoFocusChange
  Not documented.
- DoFocusChanging
  Not documented.
- DoFocusNode
  Internal method to set the focused node.
- DoFreeNode
  Not documented.
- DoGetAnimationType
  Determines the type of animation to be used.
- DoGetCursor
  Not documented.
- DoGetHeaderCursor
  Not documented.
- DoGetImageIndex
  Not documented.
- DoGetLineStyle
  Not documented.
- DoGetNodeHint
  Not documented.
- DoGetNodeTooltip
  Not documented.
**DoGetNodeWidth**
Overridable method which always returns 0.

**DoGetPopupMenu**
Overridable method which triggers the OnGetPopup event.

**DoGetUserClipboardFormats**
Not documented.

**DoHeaderClick**
Not documented.

**DoHeaderDb1Click**
Not documented.

**DoHeaderDragged**
Not documented.

**DoHeaderDraggedOut**
Not documented.

**DoHeaderDragging**
Not documented.

**DoHeaderDraw**
Not documented.

**DoHeaderDrawQueryElements**
Not documented.

**DoHeaderMouseDown**
Not documented.

**DoHeaderMouseMove**
Not documented.

**DoHeaderMouseUp**
Not documented.

**DoHotChange**
Not documented.

**DoIncrementalSearch**
Not documented.

**DoInitChildren**
Not documented.

**DoInitNode**
Not documented.

**DoKeyAction**
Not documented.

- **DoLoadUserData**
  Not documented.

- **DoMeasureItem**
  Not documented.

- **DoNodeCopied**
  Not documented.

- **DoNodeCopying**
  Not documented.

- **DoNodeMoved**
  Not documented.

- **DoNodeMoving**
  Not documented.

- **DoPaintBackground**
  Not documented.

- **DoPaintDropMark**
  Overridable method which draws the small line on top of a nodes image depending on the current drop state.

- **DoPaintNode**
  Overridable method which does nothing.

- **DoPopupMenu**
  Overridable method which shows the popup menu for the given node.

- **DoRenderOLEData**
  Not documented.

- **DoReset**
  Not documented.

- **DoSaveUserData**
  Not documented.

- **DoScroll**
  Overridable method which triggers the **OnScroll** event.

- **DoSetOffsetXY**
  Internal core routine to set the tree's scroll position.

- **DoShowScrollbar**
  Not documented.

- **DoStartDrag**
Not documented.

- **DoStateChange**
  Not documented.

- **DoStructureChange**
  Not documented.

- **DoTimerScroll**
  Callback method which is triggered whenever the scroll timer fires.

- **DoUpdating**
  Not documented.

- **DoValidateCache**
  Not documented.

- **DragCanceled**
  Called by the VCL when a drag’n drop operation was canceled by the user.

- **DragDrop**
  Helper method, which is used when a drag operation is finished.

- **DragEnter**
  Not documented.

- **DragFinished**
  Called when a drag operation is finished (accepted or cancelled).

- **Dragging**
  Returns true if a drag’n drop operation is in progress.

- **DragLeave**
  Not documented.

- **DragOver**
  Not documented.

- **DrawDottedHLine**
  Not documented.

- **DrawDottedVLine**
  Not documented.

- **EditNode**
  Starts editing the given node if allowed to.

- **EndEditNode**
  Stops node editing if it was started before.

- **EndSynch**
Counterpart to BeginSynch.

EndUpdate
Resets the update lock set by BeginUpdate.

ExecuteAction
Not documented.

FindNodeInSelection
Helper method to find the given node in the current selection.

FinishChunkHeader
Not documented.

FinishCutOrCopy
Stops any pending cut or copy clipboard operation.

FlushClipboard
Renders all pending clipboard data.

FontChanged
Not documented.

FullCollapse
Collapses all nodes in the tree.

FullExpand
Expands all nodes in the tree.

GetBorderDimensions
Not documented.

GetCheckImage
Not documented.

GetCheckImageListFor
Not documented.

GetColumnClass
Returns the class to be used to manage columns in the tree.

GetControlsAlignment
Not documented.

GetDisplayRect
Returns the visible region used by the given node in client coordinates.

GetFirst
Group of node navigation functions.

GetFirstChecked
Not documented.

- **GetFirstChild**
  Group of node navigation functions.

- **GetFirstCutCopy**
  Group of node navigation functions.

- **GetFirstInitialized**
  Group of node navigation functions.

- **GetFirstNoInit**
  Group of node navigation functions.

- **GetFirstSelected**
  Group of node navigation functions.

- **GetFirstVisible**
  Group of node navigation functions.

- **GetFirstVisibleChild**
  Group of node navigation functions.

- **GetFirstVisibleChildNoInit**
  Group of node navigation functions.

- **GetFirstVisibleNoInit**
  Group of node navigation functions.

- **GetHeaderClass**
  Returns the header class to be used by the tree.

- **GetHintWindowClass**
  Not documented.

- **GetHitTestInfoAt**
  Returns information about the node at the given position.

- **GetImageIndex**
  Not documented.

- **GetLast**
  Group of node navigation functions.

- **GetLastChild**
  Group of node navigation functions.

- **GetLastChildNoInit**
  Group of node navigation functions.

- **GetLastInitialized**
  Group of node navigation functions.
**GetLastNoInit**
Group of node navigation functions.

**GetLastVisible**
Group of node navigation functions.

**GetLastVisibleChild**
Group of node navigation functions.

**GetLastVisibleChildNoInit**
Group of node navigation functions.

**GetLastVisibleNoInit**
Group of node navigation functions.

**GetMaxColumnWidth**
Returns the width of the largest node in the given column.

**GetMaxRightExtend**
Determines the maximum width of the currently visible part of the tree.

**GetNativeClipboardFormats**
Used to let descendants and the application add their own supported clipboard formats.

**GetNext**
Group of node navigation functions.

**GetNextChecked**
Not documented.

**GetNextCutCopy**
Group of node navigation functions.

**GetNextInitialized**
Group of node navigation functions.

**GetNextNoInit**
Group of node navigation functions.

**GetNextSelected**
Group of node navigation functions.

**GetNextSibling**
Group of node navigation functions.

**GetNextVisible**
Group of node navigation functions.

**GetNextVisibleNoInit**
Group of node navigation functions.
GetNextVisibleSibling
Group of node navigation functions.
GetNextVisibleSiblingNoInit
Group of node navigation functions.
GetNodeAt
Not documented.
GetNodeData
Returns the address of the user data area of the given node.
GetNodeLevel
Returns the indentation level of the given node.
GetOptionsClass
Customization helper to determine which options class the tree should use.
GetPrevious
Group of node navigation functions.
GetPreviousInitialized
Group of node navigation functions.
GetPreviousNoInit
Group of node navigation functions.
GetPreviousSibling
Group of node navigation functions.
GetPreviousVisible
Group of node navigation functions.
GetPreviousVisibleNoInit
Group of node navigation functions.
GetPreviousVisibleSibling
Group of node navigation functions.
GetPreviousVisibleSiblingNoInit
Group of node navigation functions.
GetSortedCutCopySet
Returns a sorted list of nodes, which are marked for cut or copy clipboard operation.
GetSortedSelection
Returns a sorted list of all currently selected nodes.
GetTextInfo
Helper method for node editors, hints etc.

- GetTreeFromDataObject
  OLE drag'n drop and clipboard support method.
- GetTreeRect
  Returns the size of the virtual tree image.
- GetVisibleParent
  Returns the first (nearest) parent node, which is visible.
- HandleHotTrack
  Not documented.
- HandleIncrementalSearch
  Not documented.
- HandleMouseDb1Click
  Not documented.
- HandleMouseDown
  Not documented.
- HandleMouseUp
  Not documented.
- HasAsParent
  Determines if the given node has got another node as one of its parents.
- HasImage
  Not documented.
- HasPopupMenu
  Determines whether there is a pop up menu assigned to the tree.
- InitChildren
  Not documented.
- InitNode
  Not documented.
- InsertNode
  Inserts a new node and returns it to the caller.
- InternalAddFromStream
  Not documented.
- InternalAddToSelection
  Not documented.
- InternalCacheNode
Not documented.

**InternalClearSelection**
Not documented.

**InternalConnectNode**
Not documented.

**InternalData**
Returns the address of the internal data for a tree class.

**InternalDisconnectNode**
Not documented.

**InternalRemoveFromSelection**
Not documented.

**InvalidateCache**
Empties the internal node cache and marks it as invalid.

**InvalidateChildren**
Invalidates all children of the given node.

**InvalidateColumn**
Invalidates the client area part of a column.

**InvalidateNode**
Invalidates the given node.

**InvalidateToBottom**
Invalidates the client area starting with the top position of the given node.

**InvertSelection**
Inverts the current selection.

**IsEditing**
Tells the caller whether the tree is currently in edit mode.

**IsMouseSelecting**
Tell the caller whether the tree is currently in draw selection mode.

**IterateSubtree**
Iterator method to go through all nodes of a given sub tree.

**Loaded**
Not documented.

**LoadFromFile**
Loads previously streamed out tree data back in again.

**LoadFromStream**
Loads previously streamed out tree data back in again.

- **MainColumnChanged**
  Not documented.

- **MarkCutCopyNodes**
  Not documented.

- **MeasureItemHeight**
  Not documented.

- **MouseMove**
  Not documented.

- **MoveTo**
  Moves Source and all its child nodes to **Target**.

- **Notification**
  Not documented.

- **OriginalWMNCPaint**
  Not documented.

- **Paint**
  TControl's Paint method used here to display the tree.

- **PaintCheckImage**
  Not documented.

- **PaintImage**
  Not documented.

- **PaintNodeButton**
  Not documented.

- **PaintSelectionRectangle**
  Not documented.

- **PaintTree**
  Main paint routine for the tree image.

- **PaintTreeLines**
  Not documented.

- **PanningWindowProc**
  Not documented.

- **PasteFromClipboard**
  Inserts the content of the clipboard into the tree.

- **PrepareDragImage**
  Not documented.
Print
Not documented.

ProcessDrop
Helper method to ease OLE drag'n drop operations.

ProcessOLEData
Takes serialized OLE tree data and reconstructs the former structure.

ReadChunk
Not documented.

ReadNode
Not documented.

RedirectFontChangeEvent
Not documented.

ReinitChildren
Forces all child nodes of Node to be reinitialized.

ReinitNode
Forces a reinitialization of the given node.

RemoveFromSelection
Removes the given node from the current selection.

RenderOLEData
Renders pending OLE data.

RepaintNode
Causes the treeview to repaint the given node.

ResetNode
Resets the given node to uninitialized.

ResetRangeAnchor
Not documented.

RestoreFontChangeEvent
Not documented.

SaveToFile
Saves the entire content of the tree into a file or stream.

SaveToStream
Saves the entire content of the tree into a file or stream.

ScrollIntoView
Scrolls the tree so that the given node comes in the client area.

SelectAll
Selects all nodes in the tree.

SelectNodes
Selects a range of nodes.

SetBiDiMode
Not documented.

SetFocusedNodeAndColumn
Not documented.

SkipNode
Not documented.

Sort
Sorts the given node.

SortTree
Sorts the entire tree view.

StartWheelPanning
Not documented.

StopWheelPanning
Not documented.

StructureChange
Not documented.

SuggestDropEffect
Not documented.

ToggleNode
Changes a node's expand state to the opposite state.

ToggleSelection
Toggles the selection state of a range of nodes.

UnselectNodes
Deselects a range of nodes.

UpdateAction
Not documented.

UpdateDesigner
Not documented.

UpdateEditBounds
Not documented.

UpdateHeaderRect
Not documented.
**UpdateHorizontalScrollBar**
Applies changes to the horizontal and vertical scrollbars.

**UpdateScrollBars**
Applies changes to the horizontal and vertical scrollbars.

**UpdateVerticalScrollBar**
Applies changes to the horizontal and vertical scrollbars.

**UpdateWindowAndDragImage**
Not documented.

**UseRightToLeftReading**
Helper method for right-to-left layout.

**ValidateCache**
Initiates the validation of the internal node cache.

**ValidateChildren**
Validates all children of a given node.

**ValidateNode**
Validates a given node.

**ValidateNodeDataSize**
Helper method for node data size initialization.

**WndProc**
Redirected window procedure to do some special processing.

**WriteChunks**
Writes the core chunks for the given node to the given stream.

**WriteNode**
Writes the cover (envelop) chunk for the given node to the given stream.
**TVirtualStringTree Class Properties**

**Properties**

- **Action**
  Not documented.

- **Align**
  Not documented.

- **Alignment**
  Determines the horizontal alignment of text if no columns are defined.

- **Anchors**
  Not documented.

- **AnimationDuration**
  Determines the maximum duration the tree can use to play an animation.

- **AutoExpandDelay**
  Time delay after which a node gets expanded if it is the current drop target.

- **AutoScrollDelay**
  Time which determines when auto scrolling should start.

- **AutoScrollInterval**
  Time interval between scroll events when doing auto scroll.

- **Background**
  Holds a background image for the tree.

- **BackgroundOffsetX**
  Horizontal offset of the background image.

- **BackgroundOffsetY**
  Vertical offset of the background image.

- **BevelEdges**
  Not documented.

- **BevelInner**
Not documented.

- **BevelKind**
  Not documented.
- **BevelOuter**
  Not documented.
- **BevelWidth**
  Not documented.
- **BiDiMode**
  Not documented.
- **BorderStyle**
  Same as TForm.BorderStyle.
- **BorderWidth**
  Not documented.
- **ButtonFillMode**
  Determines how to fill the background of the node buttons.
- **ButtonStyle**
  Determines the look of node buttons.
- **Canvas**
  Not documented.
- **ChangeDelay**
  Time which determines when the OnChange event should be triggered after the actual change event.
- **CheckImageKind**
  Determines which images should be used for checkboxes and radio buttons.
- **ClipboardFormats**
  Special class to keep a list of clipboard format descriptions.
- **Color**
  Not documented.
- **Colors**
  A collection of colors used in the tree.
- **Constraints**
  Not documented.
- **Ctl3D**
  Not documented.
CustomCheckImages
Assign your own image list to get the check images you like most.

DefaultNodeHeight
Read or set the height new nodes get as initial value.

DefaultPasteMode
Read or set the value, which determines where to add pasted nodes to.

DefaultText
Not documented.

DragCursor
Not documented.

DragHeight
Read or set the vertical limit of the internal drag image.

DragImageKind
Read or set what should be shown in the drag image.

DragKind
Not documented.

DragMode
Not documented.

DragOperations
Read or set which drag operations may be allowed in the tree.

DragType
Read or set which subsystem should be used for dragging.

DragWidth
Read or set the horizontal limit of the internal drag image.

DrawSelectionMode
Read or set how multiselection with the mouse is to be visualized.

EditDelay
Read or set the maximum time between two single clicks on the same node, which should start node editing.

Enabled
Not documented.

Font
Same as TWinControl.Font.

Header
Provides access to the header instance.

- **HintAnimation**
  Read or set the current hint animation type.

- **HintMode**
  Read or set what type of hint you want for the tree view.

- **HotCursor**
  Read or set which cursor should be used for hot nodes.

- **Images**
  Read or set the tree's normal image list.

- **IncrementalSearch**
  Read or set the current incremental search mode.

- **IncrementalSearchDirection**
  Read or set the direction to be used for incremental search.

- **IncrementalSearchStart**
  Read or set where to start incremental search.

- **IncrementalSearchTimeout**
  Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

- **Indent**
  Read or set the indentation amount for node levels.

- **LineMode**
  Read or set the mode of the tree lines.

- **LineStyle**
  Read or set the mode of the tree lines.

- **Margin**
  Read or set the tree's node margin.

- **NodeAlignment**
  Read or set the node alignment value.

- **NodeDataSize**
  Read or set the extra data size for each node.

- **OnClick**
  Not documented.

- **OnDblClick**
  Not documented.

- **OnEndDock**
Not documented.
- **OnEndDrag**
  Not documented.
- **OnEnter**
  Not documented.
- **OnExit**
  Not documented.
- **OnKeyDown**
  Not documented.
- **OnKeyPress**
  Not documented.
- **OnKeyUp**
  Not documented.
- **OnMouseDown**
  Not documented.
- **OnMouseMove**
  Not documented.
- **OnMouseUp**
  Not documented.
- **OnMouseWheel**
  Not documented.
- **OnResize**
  Not documented.
- **OnStartDock**
  Not documented.
- **OnStartDrag**
  Not documented.
- **ParentBiDiMode**
  Not documented.
- **ParentColor**
  Not documented.
- **ParentCtl3D**
  Not documented.
- **ParentFont**
  Not documented.
- **ParentShowHint**
  Not documented.
- **PopupMenu**
  Not documented.
- **RootNodeCount**
  Read or set the number of nodes on the top level.
- **ScrollBarOptions**
  Reference to the scroll bar options class.
- **SelectionBlendFactor**
  Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.
- **SelectionCurveRadius**
  Read or set the current corner radius for node selection rectangles.
- **ShowHint**
  Not documented.
- **StateImages**
  Reference to the images list which is used for the state images.
- **TabOrder**
  Not documented.
- **TabStop**
  Not documented.
- **TextMargin**
  Read or set the distance of the node caption to its borders.
- **TreeOptions**
  Reference to the tree's options.
- **Visible**
  Not documented.
- **WantTabs**
  Read or set whether the tree wants to process tabs on its own.

**TCustomVirtualStringTree Class**

- **DefaultText**
  Not documented.
- **EllipsisWidth**
Not documented.

Text
Not documented.

TreeOptions
Reference to the tree's options.

TBaseVirtualTree Class

Alignment
Determines the horizontal alignment of text if no columns are defined.

AnimationDuration
Determines the maximum duration the tree can use to play an animation.

AutoExpandDelay
Time delay after which a node gets expanded if it is the current drop target.

AutoScrollDelay
Time which determines when auto scrolling should start.

AutoScrollInterval
Time interval between scroll events when doing auto scroll.

Background
Holds a background image for the tree.

BackgroundOffsetX
Horizontal offset of the background image.

BackgroundOffsetY
Vertical offset of the background image.

BorderStyle
Same as TForm.BorderStyle.

ButtonFillMode
Determines how to fill the background of the node buttons.

ButtonStyle
Determines the look of node buttons.

ChangeDelay
Time which determines when the OnChange event should be triggered after the actual change event.
CheckImageKind
Determines which images should be used for checkboxes and radio buttons.

CheckImages
Not documented.

CheckState
Read or set the check state of a node.

CheckType
Read or set the check type of a node.

ChildCount
Read or set the number of child nodes of a node.

ChildrenInitialized
Read whether a node's child count has been initialized already.

ClipboardFormats
Special class to keep a list of clipboard format descriptions.

Colors
A collection of colors used in the tree.

CustomCheckImages
Assign your own image list to get the check images you like most.

DefaultNodeHeight
Read or set the height new nodes get as initial value.

DefaultPasteMode
Read or set the value, which determines where to add pasted nodes to.

DragHeight
Read or set the vertical limit of the internal drag image.

DragImage
Holds the instance of the internal drag image.

DragImageKind
Read or set what should be shown in the drag image.

DragManager
Holds the reference to the internal drag manager.

DragOperations
Read or set which drag operations may be allowed in the tree.

DragSelection
Keeps a temporary list of nodes during drag’n drop.

- **DragType**
  Read or set which subsystem should be used for dragging.

- **DragWidth**
  Read or set the horizontal limit of the internal drag image.

- **DrawSelectionMode**
  Read or set how multiselection with the mouse is to be visualized.

- **DropTargetNode**
  Contains the current drop target node if the tree is currently the target of a drag’n drop operation.

- **EditColumn**
  Not documented.

- **EditDelay**
  Read or set the maximum time between two single clicks on the same node, which should start node editing.

- **EditLink**
  Keeps a reference to the internal edit link during a node edit operation.

- **Expanded**
  Read or set the expanded state of a particular node.

- **FocusedColumn**
  Read or set the currently focused column.

- **FocusedNode**
  Read or set the currently focused node.

- **Font**
  Same as TWinControl.Font.

- **FullyVisible**
  Read or set whether a node is fully visible or not.

- **HasChildren**
  Read or set whether a node has got children.

- **Header**
  Provides access to the header instance.

- **HeaderRect**
  Returns the non-client-area rectangle used for the header.
Read or set the current hint animation type.

**HintMode**
Read or set what type of hint you want for the tree view.

**HotCursor**
Read or set which cursor should be used for hot nodes.

**HotNode**
Read, which node is currently the hot node.

**Images**
Read or set the tree's normal image list.

**IncrementalSearch**
Read or set the current incremental search mode.

**IncrementalSearchDirection**
Read or set the direction to be used for incremental search.

**IncrementalSearchStart**
Read or set where to start incremental search.

**IncrementalSearchTimeout**
Read or set the maximum time, which is allowed between two consecutive key strokes so that incremental search stays active.

**Indent**
Read or set the indentation amount for node levels.

**IsDisabled**
Read or set the enabled state of the given node.

**IsVisible**
Read or set the visibility state of the given node.

**LastClickPos**
Used for retained drag start and wheel mouse scrolling.

**LastDropMode**
Read how the last drop operation finished.

**LineMode**
Read or set the mode of the tree lines.

**LineStyle**
Read or set the mode of the tree lines.

**Margin**
Read or set the tree's node margin.

**MultiLine**
Read or toggle the multiline feature for a given node.

- **NodeAlignment**
  Read or set the node alignment value.

- **NodeDataSize**
  Read or set the extra data size for each node.

- **NodeHeight**
  Read or set a node's height.

- **NodeParent**
  Read or set a node's parent node.

- **OffsetX**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetXY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **OffsetY**
  Read or set the tree's current horizontal and vertical scroll offsets.

- **RootNode**
  Reference to the internal root node which is the anchor of the entire tree node hierarchy.

- **RootNodeCount**
  Read or set the number of nodes on the top level.

- **ScrollBarOptions**
  Reference to the scroll bar options class.

- **SearchBuffer**
  Current input string for incremental search.

- **Selected**
  Property to modify or determine the selection state of a node.

- **SelectedCount**
  Contains the number of selected nodes.

- **SelectionBlendFactor**
  Read or set the current blend factor for the multi selection rectangle and the node selection rectangle.

- **SelectionCurveRadius**
  Read or set the current corner radius for node selection rectangles.

- **StateImages**
  Reference to the images list which is used for the state images.
TextMargin
Read or set the distance of the node caption to its borders.

TopNode
The top node is the node which is currently at the top border of the client area.

TotalCount
Returns the number of nodes in the tree.

TotalInternalDataSize
Keeps the currently accumulated data size for one node.

TreeOptions
Reference to the tree's options.

TreeStates
Property which keeps a set of flags which indicate current operation and states of the tree.

UpdateCount
Not documented.

VerticalAlignment
Used to set a node's vertical button alignment with regard to the entire node rectangle.

VisibleCount
Number of currently visible nodes.

VisiblePath
Property to set or determine a node parent's expand states.

WantTabs
Read or set whether the tree wants to process tabs on its own.
See Also
TVirtualStringTree.CustomCheckImages Property

See Also
TCheckImageKind
See Also
TVirtualStringTree.DefaultPasteMode Property

See Also
TVTNodeAttachMode
See Also
TVirtualStringTree.EditDelay Property

See Also
Editors and editing
See Also

TVirtualStringTree.Header Property

See Also

TVTHeader
See Also
TVirtualStringTree.HotCursor Property

See Also
HotNode, TVTPaintOptions
See Also
TVirtualStringTree.Images Property

See Also
StateImages, CheckImages
See Also
TVirtualStringTree.IncrementalSearch Property

See Also
IncrementalSearchDirection, IncrementalSearchStart, IncrementalSearchTimeout
See Also
TVirtualStringTree.IncrementalSearchDirection Property

See Also
IncrementalSearch, IncrementalSearchStart, IncrementalSearchTime123out
See Also
TVirtualStringTree.IncrementalSearchStart Property

See Also
IncrementalSearch, IncrementalSearchDirection, IncrementalSearchTimeout
See Also
TVirtualStringTree.IncrementalSearchTimeout Property

See Also
IncrementalSearch, IncrementalSearchDirection, IncrementalSearchStart
See Also

TVirtualStringTree.Margin Property

See Also

TVirtualStringTree.TextMargin
See Also
TVirtualStringTreeNodeAlignment Property

See Also
TVirtualNode
See Also

TVirtualStringTree.NodeDataSize Property

See Also

Data handling
See Also
TVirtualStringTree.OnAdvancedHeaderDraw Event

See Also
OnHeaderDrawQueryElements, OnHeaderDraw
See Also
TVirtualStringTree.OnAfterCellPaint Event

See Also
Paint cycles and stages
See Also
TVirtualStringTree.OnAfterItemErase Event

See Also
Paint cycles and stages
See Also
TVirtualStringTree.OnAfterItemPaint Event

See Also
Paint cycles and stages
See Also

TVirtualStringTree.OnAfterPaint Event

See Also

Paint cycles and stages
See Also

tVirtualStringTree.OnBeforeCellPaint Event

See Also

Paint cycles and stages
See Also
TVirtualStringTree.OnBeforeItemErase Event

See Also
Paint cycles and stages
See Also
TVirtualStringTree.OnBeforeItemPaint Event

See Also
Paint cycles and stages
See Also
TVirtualStringTree.OnBeforePaint Event

See Also
Paint cycles and stages
See Also
TVirtualStringTree.OnColumnClick Event

See Also
OnHeaderClick
See Also
TVirtualStringTree.OnColumnDb1Click Event

See Also
OnColumnClick, OnHeaderDb1Click
See Also

TVirtualStringTree.OnCompareNodes Event

See Also

SortTree, Sort
See Also
TVirtualStringTree.OnCreateEditor Event

See Also
Editors and editing
See Also
TVirtualStringTree.OnDragOver Event

See Also
OnDragDrop
See Also
TVirtualStringTree.OnEditCancelled Event

See Also
Editors and editing
See Also

TVirtualStringTree.OnEdited Event

See Also

Editors and editing
See Also

TVirtualStringTree.OnEditing Event

See Also

Editors and editing
See Also

TVirtualStringTree.OnGetLineStyle Event

See Also

PrepareBitmaps
See Also

TVirtualStringTree.OnGetNodeDataSize Event

See Also

NodeDataSize, Data handling
See Also
TVirtualStringTree.OnGetText Event

See Also
OnPaintText
See Also

TVirtualStringTree.OnHeaderClick Event

See Also

SortColumn, SortDirection
See Also

TVirtualStringTree.OnHeaderDoubleClick Event

See Also

OnHeaderClick
See Also

TVirtualStringTree.OnHeaderDrawQueryElements Event

See Also

OnAdvancedHeaderDraw
See Also
TVirtualStringTree.OnInitChildren Event

See Also
The virtual paradigm
See Also
TVirtualStringTree.OnInitNode Event

See Also
The virtual paradigm
See Also

TVirtualStringTree.OnLoadNode Event

See Also

OnSaveNode, LoadFromStream, SaveToStream, AddFromStream, VTTreeStreamVersion, TVTHeader.LoadFromStream, TVTHeader.SaveToStream
See Also
TvirtualStringTree.OnMeasureItem Event

See Also
InvalidateNode, vsHeightMeasured
See Also
TVirtualStringTree.OnNewText Event

See Also
OnCreateEditor, OnEdited
See Also
TVirtualStringTree.OnPaintText Event

See Also
Paint cycles and stages
See Also
TVirtualStringTree.OnResetNode Event

See Also
ResetNode
See Also

TVirtualStringTree.OnSaveNode Event

See Also

OnLoadNode, LoadFromStream, SaveToStream, AddFromStream, VTTreeStreamVersion, TVTHeader.LoadFromStream, TVTHeader.SaveToStream
See Also
TVirtualStringTree.OnScroll Event

See Also
OffsetXY
See Also
TVirtualStringTree.SelectionBlendFactor Property

See Also
DrawSelectionMode, TVTPaintOptions
See Also
TVirtualStringTree.SelectionCurveRadius Property

See Also
SelectionBlendFactor, DrawSelectionMode, TVTPaintOptions
See Also
TVirtualStringTree.StateImages Property

See Also
CheckImages, Images
See Also
TVirtualStringTree.TextMargin Property

See Also
Margin
**TVirtualTreeColumn Class Methods**

**Methods**

- **Assign**
  Not documented.
- **ComputeHeaderLayout**
  Calculates the layout of a column header.
- **Create**
  Not documented.
- **DefineProperties**
  Not documented.
- **Destroy**
  Not documented.
- **Equals**
  Not documented.
- **GetAbsoluteBounds**
  Not documented.
- **GetDisplayName**
  Not documented.
- **GetOwner**
  Not documented.
- **GetRect**
  Returns the rectangle this column occupies in the header (relative to (0, 0) of the non-client area).
- **LoadFromStream**
  Not documented.
- **ParentBiDiModeChanged**
  Not documented.
- **ParentColorChanged**
  Not documented.
ReadHint
Not documented.

ReadText
Not documented.

RestoreLastWidth
Not documented.

SaveToStream
Not documented.

UseRightToLeftReading
Not documented.

WriteHint
Not documented.

WriteText
Not documented.
TVirtualTreeColumn Class Properties

Properties

- Alignment
  Not documented.
- BiDiMode
  Not documented.
- Color
  Not documented.
- Hint
  Not documented.
- ImageIndex
  Not documented.
- Layout
  Not documented.
- Left
  Not documented.
- Margin
  Not documented.
- MaxWidth
  Not documented.
- MinWidth
  Not documented.
- Options
  Not documented.
- Owner
  Not documented.
- Position
  Not documented.
**Spacing**
Not documented.

**Style**
Not documented.

**Tag**
Not documented.

**Text**
Not documented.

**Width**
Not documented.
Legend

*TVirtualTreeColumn Class*

**Legend**

- Published
- Property
- Public
- Read only
- Method
- Virtual
- Protected
TVirtualTreeColumns Class Methods

Methods

- **Add**
  Not documented.
- **AdjustAutoSize**
  Called when columns must be sized so that they fit the client area.
- **AdjustDownColumn**
  Determines the column from the given position and returns it.
- **AdjustHoverColumn**
  Determines the new hover column index and returns true if the index actually changed else False.
- **AdjustPosition**
  Reorders the column position array so that the given column gets the given position.
- **AnimatedResize**
  Resizes the given column animated by scrolling the window DC.
- **Assign**
  Not documented.
- **Clear**
  Not documented.
- **ColumnFromPosition**
  Returns the index of the column at the given position.
- **Create**
  Not documented.
- **Destroy**
  Not documented.
- **DrawButtonText**
  Not documented.
- **DrawXPButton**
Helper procedure to draw an Windows XP like header button.

- **Equals**
  Compares itself with the given set of columns.

- **FixPositions**
  Fixes column positions after loading from DFM.

- **GetColumnAndBounds**
  Returns the column where the mouse is currently in as well as the left and right bound of this column.

- **GetColumnBounds**
  Returns the left and right bound of the given column.

- **GetFirstVisibleColumn**
  Returns the index of the first visible column or "InvalidColumn" if either no columns are defined or all columns are hidden.

- **GetLastVisibleColumn**
  Returns the index of the last visible column or "InvalidColumn" if either no columns are defined or all columns are hidden.

- **GetNextColumn**
  Returns the next column in display order. Column is the index of an item in the collection (a column).

- **GetNextVisibleColumn**
  Returns the next visible column in display order, Column is an index into the columns list.

- **GetOwner**
  Not documented.

- **GetPreviousColumn**
  Returns the previous column in display order, Column is an index into the columns list.

- **GetPreviousVisibleColumn**
  Returns the previous column in display order, Column is an index into the columns list.

- **GetVisibleColumns**
  Returns a list of all currently visible columns in actual order.

- **GetVisibleFixedWidth**
  Not documented.
(HandleClick)
Generates a click event if the mouse button has been released over the same column it was pressed first.

(IndexChanged)
Called by a column when its index in the collection changes.

(InitializePositionArray)
Ensures that the column position array contains as much entries as columns are defined.

(IsValidColumn)
Determines whether the given column is valid or not, that is, whether it is one of the current columns.

(IsValidColumn)
Not documented.

(PaintHeader)
Not documented.

(SaveToStream)
Not documented.

(TotalWidth)
Not documented.

(Update)
Not documented.

(UpdatePositions)
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.
## TVirtualTreeColumns Class Properties

### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClickIndex</td>
<td>Not documented.</td>
</tr>
<tr>
<td>Header</td>
<td>Not documented.</td>
</tr>
<tr>
<td>HeaderBitmap</td>
<td>Not documented.</td>
</tr>
<tr>
<td>Items</td>
<td>Not documented.</td>
</tr>
<tr>
<td>PositionToIndex</td>
<td>Not documented.</td>
</tr>
<tr>
<td>TrackIndex</td>
<td>Not documented.</td>
</tr>
<tr>
<td>Legend</td>
<td>TVirtualTreeColumns Class</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------</td>
</tr>
</tbody>
</table>

**Legend**

- `public`
- `Property`
- `read only`
- `protected`
- `Method`
- `virtual`
TVirtualTreeHintWindow Class Methods

Methods

- **ActivateHint**
  Not documented.
- **CalcHintRect**
  Not documented.
- **Create**
  Not documented.
- **CreateParams**
  Not documented.
- **Destroy**
  Not documented.
- **IsHintMsg**
  The VCL is a bit too generous when telling that an existing hint can be cancelled.
- **Paint**
  Not documented.
Legend

**TVirtualTreeHintWindow Class**

Legend

- 🌍 public
- 🗻 Method
- 🌿 virtual
- 🥍 protected
TVirtualTreeOptions Class Methods

TCustomVirtualTreeOptions Class

- **AssignTo**
  Used to copy this option class to another option collection.
- **Create**
  Constructor of the class.
TVirtualTreeOptions Class Properties

Properties

- **AnimationOptions**
  Options related to animations.

- **AutoOptions**
  Options related to automatic actions.

- **MiscOptions**
  Options not related to any other category.

- **PaintOptions**
  Options related to painting.

- **SelectionOptions**
  Options related to the way nodes can be selected.

TCustomVirtualTreeOptions Class

- **AnimationOptions**
  Options related to animations.

- **AutoOptions**
  Options related to automatic actions.

- **MiscOptions**
  Options not related to any other category.

- **Owner**
  Owner tree to which the property class belongs.

- **PaintOptions**
  Options related to painting.

- **SelectionOptions**
  Options related to the way nodes can be selected.
### TVTColors Class Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign</td>
<td>Not documented.</td>
</tr>
<tr>
<td>Create</td>
<td>Not documented.</td>
</tr>
</tbody>
</table>
TVTColors Class Properties

TVTColors Class | Legend

Properties

- **BorderColor**
  - Not documented.

- **DisabledColor**
  - Not documented.

- **DropMarkColor**
  - Color of the drop mark.

- **DropTargetBorderColor**
  - Not documented.

- **DropTargetColor**
  - Not documented.

- **FocusedSelectionBorderColor**
  - Not documented.

- **FocusedSelectionColor**
  - Not documented.

- **GridLineColor**
  - Not documented.

- **HeaderHotColor**
  - Not documented.

- **HotColor**
  - Not documented.

- **SelectionRectangleBlendColor**
  - Not documented.

- **SelectionRectangleBorderColor**
  - Not documented.

- **TreeLineColor**
  - Not documented.
UnfocusedSelectionBorderColor
Not documented.

UnfocusedSelectionColor
Not documented.
TVTDataObject Class Methods

Methods

- **CanonicalUnknown**
  Helper method for setting data in the IDataObject.
- **Create**
  Constructor of the class.
- **DAvise**
  Implementation of the IDataObject.DAdvise method.
- **Destroy**
  Destructor of the class.
- **DUnadvise**
  Implementation of the IDataObject.DUnAdvise method.
- **EnumDAvise**
  Implementation of the IDataObject.EnumDAvise method.
- **EnumFormatEtc**
  Implementation of the IDataObject.EnumFormatEtc method.
- **EqualFormatEtc**
  Compares two TFormatEtc structures.
- **FindFormatEtc**
  Searches the given array for a the given format.
- **FindInternalStgMedium**
  Returns a storage medium for a given clipboard format.
- **GetCanonicalFormatEtc**
  Implementation of the IDataObject.GetCanonicalFormatEtc method.
- **GetData**
  Implementation of the IDataObject.GetData method.
- **GetDataHere**
  Implementation of the IDataObject.GetDataHere method.
- **HGlobalClone**
  Helper method for **SetData**.

- **QueryGetData**
  Implementation of the **IDataObject.QueryGetData** method.

- **RenderInternalOLEData**
  Helper method to return data previously stored by **SetData**.

- **SetData**
  Implementation of the **IDataObject.SetData** method.

- **StgMediumIncRef**
  Central managing method to copy OLE data.
Properties

- ForClipboard
  Not documented.
- FormatEtcArray
  Not documented.
- InternalStgMediumArray
  Not documented.
- Owner
  Not documented.
Legend

TVTDataObject Class

Legend

- protected
- Property
- read only
- Method
- public
- virtual
See Also
TVTDataObject.GetData Method

See Also
RenderOLEData
TVTDragImage Class Methods

Methods

- **Create**
  Not documented.
- **Destroy**
  Not documented.
- **DragTo**
  Moves the drag image to a new position, which is determined from the passed point \( P \) and the previous mouse position.
- **EndDrag**
  Not documented.
- **GetDragImageRect**
  Returns the current size and position of the drag image (screen coordinates).
- **HideDragImage**
  Not documented.
- **InternalShowDragImage**
  Frequently called helper routine to actually do the blend and put it onto
- **MakeAlphaChannel**
  Not documented.
- **PrepareDrag**
  Creates all necessary structures to do alpha blended dragging using the given image.
- **RecaptureBackground**
  Notification by the drop target tree to update the background image because something in the tree has changed.
- **ShowDragImage**
  Shows the drag image after it has been hidden by HideDragImage.
WillMove
Add a summary here...
TVTDragImage Class Properties

Properties

- **ColorKey**
  - Not documented.
- **Fade**
  - Not documented.
- **MoveRestriction**
  - Not documented.
- **PostBlendBias**
  - Not documented.
- **PreBlendBias**
  - Not documented.
- **Transparency**
  - Not documented.
- **Visible**
  - Not documented.
Legend
TVTDragImage Class

Legend

- public
- Property
- read only
- Method
- virtual
- protected
TVTDragManager Class Methods

TVTDragManager Class | Legend

Methods

- Create
  Not documented.
- Destroy
  Not documented.
- DragEnter
  Not documented.
- DragLeave
  Not documented.
- DragOver
  Not documented.
- Drop
  Not documented.
- ForceDragLeave
  This method calls the drop target helper's DragLeave method to ensure it removes the drag image from screen.
- GiveFeedback
  Not documented.
- QueryContinueDrag
  Not documented.

IVTDragManager Interface

- ForceDragLeave
  Not documented.
- GetDataObject
  Not documented.
- GetDragSource
Not documented.

- GetDropTargetHelperSupported
  Not documented.

- GetIsDropTarget
  Not documented.
TVTDragManager Class Properties

IVTDragManager Interface

- DataObject
  Not documented.
- DragSource
  Not documented.
- DropTargetHelperSupported
  Not documented.
- IsDropTarget
  Not documented.
Legend
TVTDragManager Class

Legend

- `public`
- `Method`
- `virtual`
- `Property`
- `read only`
**TVTEdit Class Methods**

**TVTEdit Class** | Legend

### Methods

- ![Image](image) **AutoAdjustSize**
  Not documented.
- ![Image](image) **Create**
  Not documented.
- ![Image](image) **CreateParams**
  Not documented.
- ![Image](image) **Release**
  Not documented.
TVTEdit Class Properties

Properties

- AutoSelect
  Not documented.
- AutoSize
  Not documented.
- BorderStyle
  Not documented.
- CharCase
  Not documented.
- HideSelection
  Not documented.
- MaxLength
  Not documented.
- OEMConvert
  Not documented.
- PasswordChar
  Not documented.
TVTHeader Class Methods

Methods

- Assign
  Not documented.
- AutoFitColumns
  Not documented.
- CanWriteColumns
  Not documented.
- ChangeScale
  Not documented.
- Create
  Not documented.
- Destroy
  Not documented.
- DetermineSplitterIndex
  Tries to find the index of that column whose right border corresponds to \( P \).
- DragTo
  Moves the drag image to a new position, which is determined from the passed point \( P \) and the previous mouse position.
- GetColumnsClass
  Returns the class to be used for the actual column implementation.
- GetOwner
  Not documented.
- GetShiftState
  Not documented.
- HandleHeaderMouseMove
  Not documented.
- HandleMessage
General message handler for the header.

- **ImageListChange**
  Not documented.

- **InHeader**
  Determines whether the given point (client coordinates!) is within the header rectangle (non-client coordinates).

- **Invalidate**
  Invalidates the entire header or parts of it so they are repainted.

- **LoadFromStream**
  Restores the state of the header from the given stream.

- **PrepareDrag**
  Initializes dragging of the header, \( P \) is the current mouse position and \( \text{Start} \) the initial mouse position.

- **ReadColumns**
  Not documented.

- **RecalculateHeader**
  Initiate a recalculation of the non-client area of the owner tree.

- **RestoreColumns**
  Restores all columns to their width which they had before they have been auto fitted.

- **SaveToStream**
  Saves the complete state of the header into the provided stream.

- **UpdateMainColumn**
  Called once the load process of the owner tree is done.

- **UpdateSpringColumns**
  Not documented.

- **WriteColumns**
  Not documented.
TVTHeader Class Properties

<table>
<thead>
<tr>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoSizeIndex</td>
</tr>
<tr>
<td>Background</td>
</tr>
<tr>
<td>Columns</td>
</tr>
<tr>
<td>DragImage</td>
</tr>
<tr>
<td>Font</td>
</tr>
<tr>
<td>Height</td>
</tr>
<tr>
<td>Images</td>
</tr>
<tr>
<td>MainColumn</td>
</tr>
<tr>
<td>Options</td>
</tr>
<tr>
<td>ParentFont</td>
</tr>
<tr>
<td>PopupMenu</td>
</tr>
<tr>
<td>SortColumn</td>
</tr>
<tr>
<td>SortDirection</td>
</tr>
</tbody>
</table>

Not documented.
States
Not documented.

Style
Not documented.

Treeview
Not documented.

UseColumns
Not documented.
Legend
TVTHeader Class

Legend

- published
- Property
- public
- read only
- Method
- virtual
- protected
TVTHeaderPopupMenu Class Events

Events

- **OnAddHeaderPopupItem**
  Not documented.

- **OnColumnChange**
  Not documented.
TVTHeaderPopupMenu Class Methods

Methods

- **DoAddHeaderPopupItem**
  Not documented.
- **DoColumnChange**
  Not documented.
- **OnMenuItemClick**
  Not documented.
- **Popup**
  Not documented.
### TVTHeaderPopupMenu Class Properties

<table>
<thead>
<tr>
<th>TVTHeaderPopupMenu Class</th>
<th>Legend</th>
</tr>
</thead>
</table>

**Properties**

- **Options**
  - Not documented.
Legend
TVTHeaderPopupMenu Class

Legend
- published
- Event
- Property
- protected
- Method
- virtual
- public
TWideBufferedString Class Methods

**Methods**

- **Add**
  - Not documented.
- **AddNewLine**
  - Not documented.
- **Destroy**
  - Not documented.
Properties

- **AsString**
  Not documented.
Legend

TWideBufferedString Class

Legend

- public
- Property
- read only
- Method
- virtual
TWorkerThread Class Methods

Methods

- AddTree
  Not documented.
- ChangeTreeStates
  Not documented.
- Create
  Not documented.
- Destroy
  Not documented.
- Execute
  Not documented.
- RemoveTree
  Not documented.
TWorkerThread Class Properties

Properties

- CurrentTree
  Not documented.
Legend

TWorkerThread Class

Legend

- public
- Property
- read only
- Method
- protected
- virtual
Functions

AlphaBlend
  General purpose procedure to blend one bitmap to another.

DrawTextW
  Paint support procedure.

EnumerateVTClipboardFormats
  Not documented.

EnumerateVTClipboardFormats
  Not documented.

GetVTClipboardFormatDescription
  Not documented.

PrtStretchDrawDIB
  Not documented.

RegisterVTClipboardFormat
  Methods to register a certain clipboard format for a given tree class.

RegisterVTClipboardFormat
  Methods to register a certain clipboard format for a given tree class.

ShortenString
  General purpose routine to shorten a Unicode string to a given maximum size.

TreeFromNode
  General purpose routine to get the tree to which a node belongs.
Structs and Records Enumerations

Enumerations

🌟 TVTTooltipLineBreakStyle
Not documented.
Structs and Records

Records

- **TBaseChunk**
  Not documented.
- **TBaseChunkBody**
  Not documented.
- **TCacheEntry**
  Not documented.
- **TChunkHeader**
  Not documented.
- **TClipboardFormatEntry**
  Not documented.
- **TClipboardFormatListEntry**
  Not documented.
- **THeaderPaintInfo**
  Not documented.
- **THitInfo**
  Not documented.
- **TInternalStgMedium**
  Not documented.
- **TRealWMNCPaint**
  Not documented.
- **TSHDragImage**
  Not documented.
- **TToggleAnimationData**
  Not documented.
- **TVirtualNode**
  Not documented.
TVTHintData
Not documented.

TVTImageInfo
Not documented.

TVTPaintInfo
Not documented.

TVTRefERENCE
Not documented.

TWMPrint
Not documented.
Legend

Structs and Records

Legend

Struct
Types Enumerations

Enumerations

- **TAddPopupItemType**
  Not documented.

- **TBlendMode**
  Not documented.

- **TChangeReason**
  Not documented.

- **TCheckImageKind**
  Determines which images should be used for checkboxes and radio buttons.

- **TCheckState**
  Returns the current state of a node's check box, radio button or node button.

- **TCheckType**
  Not documented.

- **TDragOperation**
  Not documented.

- **TDropMode**
  Not documented.

- **THeaderState**
  Not documented.

- **THintAnimationType**
  Not documented.

- **THitPosition**
  Not documented.

- **TItemEraseAction**
  Not documented.

- **TScrollBarStyle**
Not documented.

TSortDirection
Not documented.

TVirtualNodeInitState
Not documented.

TVirtualNodeState
Not documented.

TVirtualTreeColumnStyle
Not documented.

TVSTTextSourceType
Not documented.

TVSTTextType
Not documented.

TVTAnimationOption
Not documented.

TVTAutoOption
Not documented.

TVTButtonFillMode
Determines how the interior of nodes buttons should be drawn.

TVTButtonStyle
Not documented.

TVTColumnOption
Not documented.

TVTDragImageKind
Not documented.

TVTDragMoveRestriction
Not documented.

TVTDragType
Not documented.

TVTDrawSelectionMode
Not documented.

TVTDropMarkMode
Not documented.

TVTHeaderColumnLayout
Not documented.
TVTHeaderOption
Not documented.

TVTHeaderPopupOption
Not documented.

TVTHeaderStyle
Not documented.

TVTHintMode
Not documented.

TVTImageInfoIndex
Not documented.

TVTImageKind
Not documented.

TVTIncrementalSearch
Not documented.

TVTInternalPaintOption
Not documented.

TVTLineMode
Not documented.

TVTLineStyle
Not documented.

TVTLineType
Not documented.

TVTMiscOption
Not documented.

TVTNodeAlignment
Not documented.

TVTNodeAttachMode
Not documented.

TVTPaintOption
Not documented.

TVTSearchDirection
Not documented.

TVTSearchStart
Not documented.

TVTSelectionOption
Not documented.

TVTStringOption
Not documented.

TVTUpdateState
Not documented.
Types

PCardinal
Not documented.

PClipboardFormatListEntry
Not documented.

PSHDragImage
Not documented.

PVirtualNode
Not documented.

PVTHintData
Not documented.

PVTReference
Not documented.

TAddHeaderPopupItemEvent
Not documented.

TAutoScrollInterval
Not documented.

TCache
Not documented.

TCardinalArray
Not documented.

TChangeStates
Not documented.

TColumnChangeEvent
Not documented.

TColumnIndex
Not documented.
*TColumnPosition*
Not documented.

*TColumnsArray*
Not documented.

*TDragOperations*
Not documented.

*TFormatArray*
Not documented.

*TFormatEtcArray*
Not documented.

*TGetFirstNodeProc*
Not documented.

*TGetNextNodeProc*
Not documented.

*THeaderPaintElements*
Not documented.

*THeaderStates*
Not documented.

*THitPositions*
Not documented.

*TMagicID*
Not documented.

*TIndexArray*
Not documented.

*TImageIndex*
Not documented.

*TLineImage*
Not documented.

*TMagicID*
Not documented.

*TMouseButtons*
Not documented.

*TNodeArray*
Not documented.

*TScrollDirections*
Not documented.

TScrollUpdateOptions
Not documented.

TTreeOptionsClass
Not documented.

TVirtualNodeInitStates
Not documented.

TVirtualNodeStates
Not documented.

TVirtualTreeClass
Not documented.

TVirtualTreeColumnClass
Not documented.

TVirtualTreeColumnsClass
Not documented.

TVirtualTreeStates
Not documented.

TVSTGetTextEvent
Not documented.

TVSTNewTextEvent
Not documented.

TVSTShortenStringEvent
Not documented.

TVTAdvancedHeaderPaintEvent
Not documented.

TVTAfterCellPaintEvent
Not documented.

TVTAfterItemEraseEvent
Not documented.

TVTAfterItemPaintEvent
Not documented.

TVTAnimationCallback
Not documented.

TVTAnimationOptions
Not documented.
TVTAutoOptions
Not documented.

TVTBackgroundPaintEvent
Not documented.

TVTBeforeCellPaintEvent
Not documented.

TVTBeforeItemEraseEvent
Not documented.

TVTBeforeItemPaintEvent
Not documented.

TVTBias
Not documented.

TVTChangeListener
Not documented.

TVTChangingEvent
Not documented.

TVTCheckChangingEvent
Not documented.

TVTColumnClickEvent
Not documented.

TVTColumnDblClickEvent
Not documented.

TVTColumnOptions
Not documented.

TVTCompareEvent
Not documented.

TVTCreateDataObjectEvent
Not documented.

TVTCreateDragManagerEvent
Not documented.

TVTCreateEditorEvent
Not documented.

TVTDragAllowedEvent
Not documented.

TVTDragDropEvent
Not documented.

TVTDragImageStates
Not documented.

TVTDragOverEvent
Not documented.

TVTDrawHintEvent
Not documented.

TVTDrawNodeEvent
Not documented.

TVTEditCancelEvent
Not documented.

TVTEditChangeEvent
Not documented.

TVTEditChangingEvent
Not documented.

TVTFocusChangeEvent
Not documented.

TVTFocusChangingEvent
Not documented.

TVTFreeNodeEvent
Not documented.

TVTGetCursorEvent
Not documented.

TVTGetHeaderCursorEvent
Not documented.

TVTGetHintSizeEvent
Not documented.

TVTGetImageEvent
Not documented.

TVTGetLineStyleEvent
Not documented.

TVTGetNodeDataSizeEvent
Not documented.

TVTGetNodeProc
Not documented.
TVTGetNodeWidthEvent
Not documented.

TVTGetUserClipboardFormatsEvent
Not documented.

TVTHeaderClass
Not documented.

TVTHeaderClickEvent
Not documented.

TVTHeaderDraggedEvent
Not documented.

TVTHeaderDraggedOutEvent
Not documented.

TVSTGetHintEvent
Not documented.

TVTHeaderDraggingEvent
Not documented.

TVTHeaderMouseEvent
Not documented.

TVTHeaderMouseMoveEvent
Not documented.

TVTHeaderNotifyEvent
Not documented.

TVTHeaderOptions
Not documented.

TVTHeaderPaintEvent
Not documented.

TVTHeaderPaintQueryElementsEvent
Not documented.

TVTHeaderPopupOptions
Not documented.

TVTHelpContextEvent
Not documented.

TVTHotNodeChangeEvent
Not documented.

TVTIncrementalSearchEvent
Not documented.

- **TVTInitChildrenEvent**
  - Not documented.
- **TVTInitNodeEvent**
  - Not documented.
- **TVTInternalPaintOptions**
  - Not documented.
- **TVTKeyActionEvent**
  - Not documented.
- **TVTMeasureItemEvent**
  - Not documented.
- **TVTMiscOptions**
  - Not documented.
- **TVTNodeCopiedEvent**
  - Not documented.
- **TVTNodeCopyingEvent**
  - Not documented.
- **TVTNodeMovedEvent**
  - Not documented.
- **TVTNodeMovingEvent**
  - Not documented.
- **TVTPaintEvent**
  - Not documented.
- **TVTPaintOptions**
  - Not documented.
- **TVTPaintText**
  - Not documented.
- **TVTPopupEvent**
  - Not documented.
- **TVTRenderOLEDataEvent**
  - Not documented.
- **TVTSaveNodeEvent**
  - Not documented.
- **TVTScrollEvent**
  - Not documented.
TVTScrollIncrement
Not documented.

TVTSelectionOptions
Not documented.

TVTStateChangeEvent
Not documented.

TVTStringOptions
Not documented.

TVTStructureChangeEvent
Not documented.

TVTTransparency
Not documented.

TVTUpdatingEvent
Not documented.

TWMContextMenu
Not documented.

TWMPrintClient
Not documented.

TVTGetCellIsEmptyEvent
Not documented.

TVTGetImageExEvent
Not documented.

TVTMenuItem
Not documented.

TVTScrollbarShowEvent
Not documented.
Variables

<table>
<thead>
<tr>
<th>Variables</th>
<th>Legend</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF_CSV</td>
<td>Not documented.</td>
</tr>
<tr>
<td>CF_HTML</td>
<td>Not documented.</td>
</tr>
<tr>
<td>CF_VIRTUALTREE</td>
<td>Not documented.</td>
</tr>
<tr>
<td>CF_VRTF</td>
<td>Not documented.</td>
</tr>
<tr>
<td>CF_VRTFNOOBS</td>
<td>Not documented.</td>
</tr>
<tr>
<td>CF_VTREFERENCE</td>
<td>Not documented.</td>
</tr>
<tr>
<td>ClipboardDescriptions</td>
<td>Not documented.</td>
</tr>
<tr>
<td>DarkCheckImages</td>
<td>Not documented.</td>
</tr>
<tr>
<td>DarkTickImages</td>
<td>Not documented.</td>
</tr>
<tr>
<td>FlatImages</td>
<td>Not documented.</td>
</tr>
<tr>
<td>HintFont</td>
<td>Not documented.</td>
</tr>
<tr>
<td>HintWindowDestroyed</td>
<td>Not documented.</td>
</tr>
<tr>
<td>Initialized</td>
<td>Not documented.</td>
</tr>
</tbody>
</table>
- InternalClipboardFormats
  Not documented.
- IsWin2K
  Not documented.
- IsWinNT
  Not documented.
- IsWinXP
  Not documented.
- LightCheckImages
  Not documented.
- LightTickImages
  Not documented.
- MMXAvailable
  Not documented.
- NeedToUninitialize
  Not documented.
- StandardOLEFormat
  Not documented.
- SystemCheckImages
  Not documented.
- SystemFlatCheckImages
  Not documented.
- UtilityImages
  Not documented.
- Watcher
  Not documented.
- WorkerThread
  Not documented.
- WorkEvent
  Not documented.
- XPImages
  Not documented.
 Constants

AlignmentToDrawFlag
Not documented.

AllocIncrement
Not documented.

BaseChunk
Not documented.

CacheThreshold
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
Not documented.

CFSTR_CSV
Contains the registration string for certain clipboard formats.

CFSTR_HTML
Contains the registration string for certain clipboard formats.

CFSTR_RTF
Contains the registration string for certain clipboard formats.

CFSTR_RTFNOOBS
Contains the registration string for certain clipboard formats.

CFSTR_VIRTUALTREE
Contains the registration string for certain clipboard formats.

CFSTR_VTREFERENCE
Contains the registration string for certain clipboard formats.

ChangeTimer
Not documented.

ckButtonDisabled
ckButtonHot
ckButtonNormal
ckButtonPressed
ckCheckCheckedDisabled
ckCheckCheckedHot
ckCheckCheckedNormal
ckCheckCheckedPressed
ckCheckMixedDisabled
ckCheckMixedHot
ckCheckMixedNormal
ckCheckMixedPressed
ckCheckUncheckedDisabled
ckCheckUncheckedHot
ckCheckUncheckedNormal
ckCheckUncheckedPressed
ckEmpty
ckRadioCheckedDisabled
ckRadioCheckedHot
ckRadioCheckedNormal
ckRadioCheckedPressed
ckRadioUncheckedDisabled
ckRadioUncheckedHot
ckRadioUncheckedNormal
ckRadioUncheckedPressed
ClipboardStates
   Not documented.
CLSID_DragDropHelper
   Not documented.
CM_AUTOADJUST
   Not documented.
CM_DENYSUBCLASSING
   Not documented.
Copyright
   Not documented.
- `crHeaderSplit`
  Not documented.
- `DefaultAnimationOptions`
  Not documented.
- `DefaultAutoOptions`
  Not documented.
- `DefaultColumnOptions`
  Not documented.
- `DefaultMiscOptions`
  Not documented.
- `DefaultPaintOptions`
  Not documented.
- `DefaultScrollUpdateFlags`
  Not documented.
- `DefaultSelectionOptions`
  Not documented.
- `DefaultStringOptions`
  Not documented.
- `EditTimer`
  Not documented.
- `ExpandTimer`
  Not documented.
- `FadeAnimationStepCount`
  Not documented.
- `Grays`
  Not documented.
- `hcTFCannotSetUserData`
  Not documented.
- `hcTFClipboardFailed`
  Not documented.
- `hcTFCorruptStream1`
  Not documented.
- `hcTFCorruptStream2`
  Not documented.
- `hcTFEditLinkIsNil`
Not documented.

hcTFStreamTooSmall
Not documented.

hcTFWrongMoveError
Not documented.

hcTFWrongStreamFormat
Not documented.

hcTFWrongStreamVersion
Not documented.

HeaderTimer
Not documented.

IID_IDragSourceHelper
Not documented.

IID_IDropTarget
Not documented.

IID_IDropTargetHelper
Not documented.

InvalidColumn
Not documented.

MagicID
Not documented.

MinimumTimerInterval
Not documented.

MouseButtonDown
Not documented.

NoColumn
Not documented.

NodeChunk
Not documented.

OptionMap
Not documented.

PressedState
Not documented.

RTLFlag
Not documented.
SCannotSetUserData
Not documented.
SClipboardFailed
Not documented.
SCorruptStream1
Not documented.
SCorruptStream2
Not documented.
ScrollTimer
Not documented.
SearchTimer
Not documented.
SEditLinkIsNil
Not documented.
ShadowSize
Size in pixels of the hint shadow.
SID_IDragSourceHelper
Not documented.
SID_IDropTarget
Not documented.
SID_IDropTargetHelper
Not documented.
SStreamTooSmall
Not documented.
StructureChangeTimer
Not documented.
SWrongMoveError
Not documented.
SWrongStreamFormat
Not documented.
SWrongStreamVersion
Not documented.
SysGrays
Not documented.
TreeNodeSize
Not documented.

UnpressedState
Not documented.

UserChunk
Not documented.

UtilityImageSize
Not documented.

VTHdrerStreamVersion
Not documented.

VTTreeStreamVersion
Not documented.

VTVersion
Not documented.

WideCR
Not documented.

WideLF
Not documented.

WideLineSeparator
Not documented.

WideNull
Not documented.

WM_CHANGETSTATE
Not documented.

XPDarkGradientColor
Not documented.

XPDarkSplitBarColor
Not documented.

XPDownInnerLineColor
Not documented.

XPDownMiddleLineColor
Not documented.

XPDownOuterLineColor
Not documented.

XPLightSplitBarColor
Not documented.
XPMainHeaderColorDown
Not documented.

XPMainHeaderColorHover
Not documented.

XPMainHeaderColorUp
Not documented.
Topics

Constants

Topics

Check button image indices
Symbol Reference Interfaces

Interases

- IDragSourceHelper
  Not documented.
- IDropTargetHelper
  Not documented.
- IVTDragManager
  Not documented.
- IVTEditLink
  Interface which is used for communication between the treeview and a node editor.
Legend

Symbol Reference

Legend

Class
IDragSourceHelper Interface Methods
IDragSourceHelper Interface | Legend

Methods

- **InitializeFromBitmap**
  - Not documented.
- **InitializeFromWindow**
  - Not documented.
Legend
IDragSourceHelper Interface

Legend

public
Method
IDropTargetHelper Interface Methods

IDropTargetHelper Interface | Legend

Methods

- DragEnter
  Not documented.
- DragLeave
  Not documented.
- DragOver
  Not documented.
- Drop
  Not documented.
- Show
  Not documented.
Legend

IDropTargetHelper Interface

Legend

- public
- Method
IVTDragManager Interface Methods

IVTDragManager Interface | Legend

Methods

- **ForceDragLeave**
  Not documented.

- **GetDataObject**
  Not documented.

- **GetDragSource**
  Not documented.

- **GetDropTargetHelperSupported**
  Not documented.

- **GetIsDropTarget**
  Not documented.
Properties

- **DataObject**
  Not documented.
- **DragSource**
  Not documented.
- **DropTargetHelperSupported**
  Not documented.
- **IsDropTarget**
  Not documented.
Legend

IVTDragManager Interface

Legend

- public
- Property
- read only
- Method
IVTEditLink Interface Methods

Methods

- **BeginEdit**
  This function will be called by the virtual tree when the editing starts.

- **CancelEdit**
  This function will be called by the virtual tree when the current editing is about to be cancelled.

- **EndEdit**
  This function will be called by the virtual tree when the current editing is being finished.

- **GetBounds**
  The virtual tree can use this function to get the current bounding rect of the node editor.

- **PrepareEdit**
  This function is called by a virtual tree to initialize the node editor.

- **ProcessMessage**
  This function is used to forward messages being directed to the virtual tree.

- **SetBounds**
  The virtual tree calls this function to initialize the bounding rectangle of the node editor.
### Legend

**IVTEditLink Interface**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td>Method</td>
</tr>
</tbody>
</table>

---
Legend

Class
Legend

- protected
- Event
Legend

- public
- Method
- protected
- virtual
Legend

- protected
- Property
- public
- read only
Legend

- public
- Property
- read only
Legend

- public
- Method
- virtual
Legend

- public
- Method
- virtual
Legend

- public
- Property
- read only
Legend

- protected
- Data Member
Legend

- public
- Method
- virtual
Legend

- public
- Method
- virtual
Legend

- protected
- Property
- public
- read only
Legend

- protected
- Event
Legend

- protected
- Method
- virtual
- public
Legend

- protected
- Property
- public
- read only
Legend

- protected
- Event
Legend

- protected
- Method
- virtual
- public
Legend

- protected
- Property
- read only
- public
Legend

- public
- Method
- virtual
Legend

- protected
- Property
- public
- read only
Legend

public

Method
Legend

- public
- Method
- virtual
- protected
Legend

- published
- Property
Legend

- public
- Method
- virtual
Legend

- public
- Property
Legend

- public
- Method
- virtual
Legend

- published
- Property
- protected
- public
- read only
Legend

- published
- Event
- protected
Legend

- protected
- Method
- virtual
- public
Legend

- published
- Property
- public
- protected
- read only
Legend

- published
- Event
- protected
Legend

- protected
- Method
- virtual
- public
Legend

- published
- Property
- public
- protected
- read only
Legend

- published
- Property
- public
- read only
Legend

- public
- Property
- read only
- protected
Legend

- public
- Method
- virtual
- protected
Legend

- public
- Method
- virtual
Legend

public
Method
virtual
Legend

- published
- Property
Legend

- protected
- Method
- public
- virtual
Legend

- protected
- Property
- read only
Legend

- public
- Method
- virtual
- protected
Legend

- public
- Property
- read only
Legend

- public
- Method
- virtual
Legend

- public
- Property
- read only
Legend

- protected
- Method
- public
- virtual
Legend

- public
- Property
Legend

- public
- Method
- virtual
- protected
Legend

- published
- Property
- public
- read only
Legend

- **published**
- **Event**
Legend

- protected
- Method
- virtual
- public
Legend

- published
- Property
Legend

- public
- Method
- virtual
Legend

- public
- Property
- read only
Legend

- public
- Method
- protected
- virtual
Legend

- public
- Property
- read only
Legend

Struct
Legend

Struct
Legend

Struct
Legend

Type
Legend

Variable
Legend

Constant
Legend

Class
Legend

public

Method
Legend

- public
- Method
Legend

public

Method
Legend

- public
- Property
- read only