

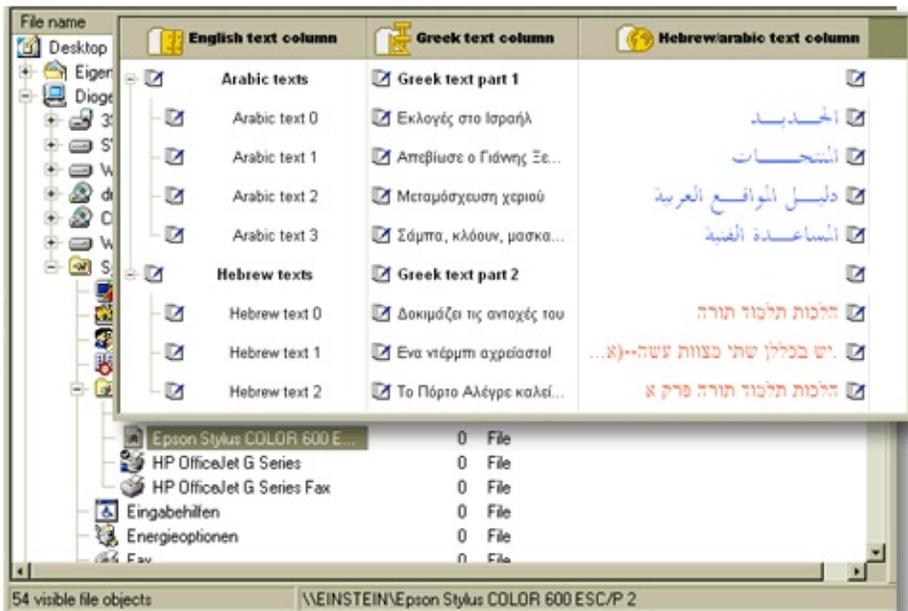


Introduction

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.

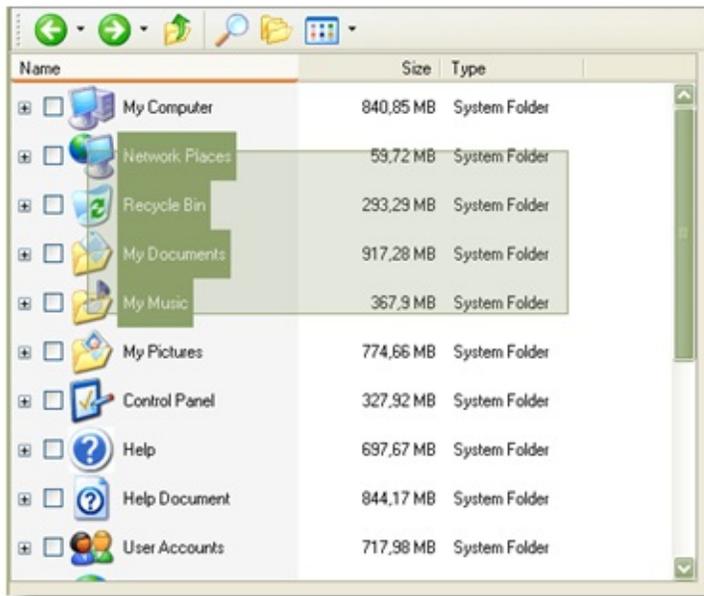
As the name already indicates, this control uses a different paradigm for tree management than other controls of this kind. It does not know anything about the data it manages (except its size), not even the captions of a node. Everything is retrieved from the application via events (or descendants via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

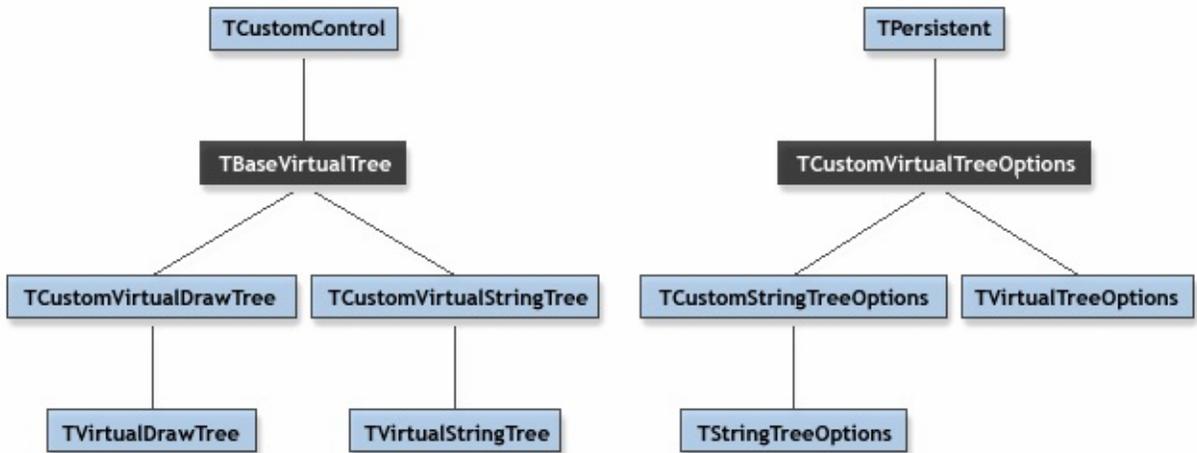
Web based forums: support.soft-gems.net/forums

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

What do you think about this topic? [Send feedback!](#)



Features overview



Virtual Treeview base hierarchy

Virtual Tree Options hierarchy

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



General

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

window). A special cursor can be assigned for this task.

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



Newest technologies

- For **smooth animations** (e.g. hint fading) Virtual Treeview uses **hand optimized MMX assembler** routines. This code is also used to draw the translucent selection rectangle in multi selection mode. This is very much like what Windows 2000 and Windows XP support but works also on Windows 95/98/Me.
- An **alpha blended image** of the tree window is shown while doing drag and drop. On Windows 2000 and Windows XP **IDropTargetHelper** 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

support can be enabled via a compiler switch named **TntSupport**. You must download and install the TNT package first, however.



Drag'n drop and clipboard support

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

These formats are support by the standard implementation:

- **Native serialized format** (**CF_VIRTUALTREE** 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 powerfull idea too. **Auto expand** of nodes which are the drop target for more than an adjustable time interval is performed if enabled.



Header and columns

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

treeview streaming.

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



Check support

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



Design time

- Virtual Treeview's properties and methods are registered with

Delphi categories (Delphi 5 and BCB 5 or higher).

- A special **property editor** for the **clipboard formats** is included which allows a simple format choice. This is particularly important since the available clipboard formats must be given as strings and it is also quite handy to have a list of available formats, even if they are not enabled yet (to know what can be enabled).



Customization

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

*

- Applications and descendants can provide their **own drag manager interface** by handling the
- OnCreateDragManager event or overriding DoCreateDragManager. This allows to customize the entire OLE drag handling of the
- tree. Note: VCL drag'n drop is managed by the VCL so this cannot

be customized.

- Applications and descendants can provide their **own data object interface** by handling the OnCreateDataObject event or overriding DoCreateDataObject. This allows to provide own clipboard formats.
- There is registration function ([RegisterVTClipboardFormat](#)) 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 **powerfull mechanism** for descendants writers to allocate their **own data** on a **per node** basis. Simply call `AllocateInternalData` to register your needs. This will not influence existing or future application code if it consequently uses `GetNodeData` for user data access.



Editing

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



Utilities

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

- **AlphaBlend**: 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 Windows 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.

VirtualTreesC4.bpk, VirtualTreesC4.cpp, VirtualTreesC4.res).

VirtualTreesReg.pas

Registration unit for some property editors and categories.

VTHederPopup.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 **NeedToUnitalize** 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 OnGetCellsEmpty
- Improvement: **TVTColors.HeaderHotColor** introduced, default value is clBtnShadow as it was hard wired before.
- Improvement: Auto spring feature. Size changes of the header are evenly spread over all columns, which are enabled for this feature. New options introduced: coAutoSpring, hoAutoSpring.
- Change: Header stream version increased to 3. This was necessary because the new coAutoSpring options increased a column's option size from byte to word (now there are 9 options).
- Improvement: Edit property of **TStringEditLink** 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 `TVTHeader.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 safe.
- 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 VTHHeaderPopup.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: **TVTHHeader.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.
- **TVTHHeader.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 explicitly 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 checkimages 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

- HasPopupMenu
- other small changes
- bug fixes

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

- Release candidate 2 for the beta testers and early adopters
- Full Windows XP theme support
- Legacy code included for XP style support on non-XP systems ([TCheckImageKind](#), [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_RTTF* to CF_VRTTF* to avoid identifier conflicts
- internal data handling improved, method AllocateInternalDataArea, method InternalData
- improved text painting
- rounded selection rectangles, property SelectionCurveRadius
- selection border colors
- hatSystemDefault, DoGetAnimationType
- small changes
- bug fixes

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

- introduced build numbers
- Delphi 6 compatibility
- brush alignment bug for dotted lines fixed
- test for TYMED_HGLOBAL is now done using a mask instead of direct comparation
- tree column classes can now be changed by descendants, see

TBaseVirtualTree.GetColumnClass

- 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
- TVTDataObject, 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, tsOLEDragging
- 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.)
- hoDbtClickResize
- **TVirtualTreeColumns.AnimatedResize**
- column resize on double click
- GetMaxColumnWidth
- poDrawFocusRect, poDrawSelection in paint options
- ChildNodesOnly in IterateSubtree
- OnColumnClick, OnColumnDbtClick
- HandleMouseDbtClick, WM_RBUTTONDOWNBLCLK, WM_MBUTTONDOWNBLCLK
- 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 (OnHeaderDbtClick, OnHeaderMouseDown, OnHeaderMouseMove, OnHeaderMouseUp)
- virtual event trigger methods for those mouse events
- multiline hints
- **THitPositions**
- other small changes

- bug fixes

Version 2.0 to 2.1 (January to February 2001)

- improved hinting (accounts now for alignment and directionality)
- improved GetDisplayRect
- FNodeCache removed
- BidiMode in OnDrawNode
- DetermineHitPositionLTR, DetermineHitPositionRTL
- improved GetHitTestInfoAt
- improved GetNodeAt method
- FindInPositionCache
- made the header the sender in all events related to the header (e.g. OnHeaderClick)
- WM_PRINT, WM_PRINTCLIENT
- Text property for **TVirtualStringTree**
- 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)
- volnitOnSave
- 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/collapes on node focuse change, DeleteSelectedNodes, SelectAll, InvertSelection, GetNext(Previous)Sibling
- splitted change event into two, one for node focus change and one for structure change (Add, Delete etc.)
- disabled nodes, disabled tree, cutted nodes (with proper visual feedback)
- column images
- hideable columns, columns auto sizing and reordering
- normal and static text for captions
- general optimizations
- application driven help and popup menu
- compatibility for Delphi 4
- bug fixes

Versions 1.2 to 1.3 (January 2000)

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

Version 1.1 (December 1999)

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

Version 1.0 (July 1999 to November 1999)

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

What do you think about this topic? [Send feedback!](#)



The virtual paradigm

Inner fundamentals

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

Description

The History

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

The problems

Despite the problems with the system tree control TreeNT worked quite well and has meanwhile been downloaded several thousands of times from my web site and those many other Delphi sites around the world. When I started working

for a software house in Munich I quickly included TreeNT into the company's inhouse library. But then the problems which were formerly only annoying started to make the tree nearly unusable. I realized how much the requirements in the private and professional/commercial environment actually differ.

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

The nodes

These thoughts inspired the idea of using small records as nodes only and putting them into a doubly linked list (see also [TVirtualNode](#)). 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 rescan and update of the tree. Since this would be much too expensive a node cache has been introduced. This cache is a simple one-dimensional array which holds node references in increasing absolute position order. A separate thread (which is shared between all Virtual Treeview instances in a program) is used to collect the references in the background. Well, one could say that all these updates are still necessary (even with a cache because it must be held coherent) and the thread could well work

directly in the node records. The most valuable advantage of the array like cache is however that you can query it for a node at a particular position by using binary search which is not possible with linked lists.

The paradigm

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

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

updates for the tree implicitly unless updates are locked. It is therefore strongly recommended to put calls to AddChild and InsertNode always into a BeginUpdate/EndUpdate frame (if there is more than one call).

Records instead classes

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

With choosing records I also gave up the VCL concept of having a tree nodes class which is responsible to manage tree nodes and is secondary to the control itself. In Virtual

Treeview every access to the tree content is done via methods and properties provided by the tree control. Keep also in mind that nobody prevents you from using classes and store their references in the node's data area. It is only just so that the node (as internal management structure) is as small as possible, opening so all possibilities: from smallest memory footprint to highest comfort.

19.09.2003

Times are changing

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

Group

Inner fundamentals

Links

Inner fundamentals

What do you think about this topic? [Send feedback!](#)



Paint cycles and stages

Inner fundamentals

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

Description

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

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

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

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

after item paint (OnAfterItemPaint) after paint (OnAfterPaint)

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

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

Stage	Description	Comments
	This stage is entered only once per paint cycle. After	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

before paint	setting the vsPainting state it is the very first instruction in a cycle.	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.
before item paint	This stage is entered once per node to be drawn and allows directly to control the path which is taken to paint the node.	<p>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. 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 the event to True will skip the node entirely without painting anything of the standard things like tree lines, buttons, images or erasing the background. Hence to display any useful information for the node do it in the OnBeforeItemPaint event.</p> <p>This is the first stage which gets the double buffer canvas which is used to draw a node so if you want to set special properties this is a good opportunity. Keep in mind though in particular the colors are set by the tree according to specific rules (focus, selection etc.).</p>
before item erase	This stage is also entered only once per node and allows to customize the node's background.	This stage and its associated event is used to give the node a different background color or erase the background with a specific pattern which is different to what the tree would draw.
after	This stage is also entered	This stage and its associated event is used

item erase	only once per node.	to do additional drawings after the background has been erased.
before cell paint	This paint stage is the first of the cell specific stages used to customize a single cell of a node and is called several times per node, depending on the number of columns. If no columns are used then it is called once.	While internally a full setup for this node happened before the stage is entered (if i the first run) the only noticeable effect for application which has changed comparac after item erase is that the painting is lim to the current column. There are still no li or images painted yet.
on paint text	After default stuff like lines and images has been painted the paint node/paint text stage is entered.	Because Virtual Treeview does not know 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. I instance TVirtualDrawTree simply trigger: OnDrawNode event while the TVirtualStringTree prepares the target canvas and allows the application to over 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 t text. Note: The string tree triggers the OnGetI event two times if toShowStaticText is enabled in the TVirtualStringTree.TreeOptions.StringOp

		property. Once for the normal text and once for the static text. Use the event's parameters to find out what is required.
after cell paint	This stage is entered immediately after the cell is drawn.	This stage can be used to add whatever you want like to a single cell after everything has been painted there and is triggered once per column.
after item paint	This stage is entered after all cells of an item are drawn.	The after item paint stage is used to add node specific stuff like frames and the like which concern all columns of that node and is called once per node.
after paint	The after paint stage is the last stage in the long chain of paint stages and is entered after when paint cycle is complete.	In this stage everything of the tree (relative to the current update area) has been drawn including the selection rectangle.

Group

[Inner fundamentals](#)

Links

[Inner fundamentals](#)

What do you think about this topic? [Send feedback!](#)



Tree image and tree window

Inner fundamentals

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

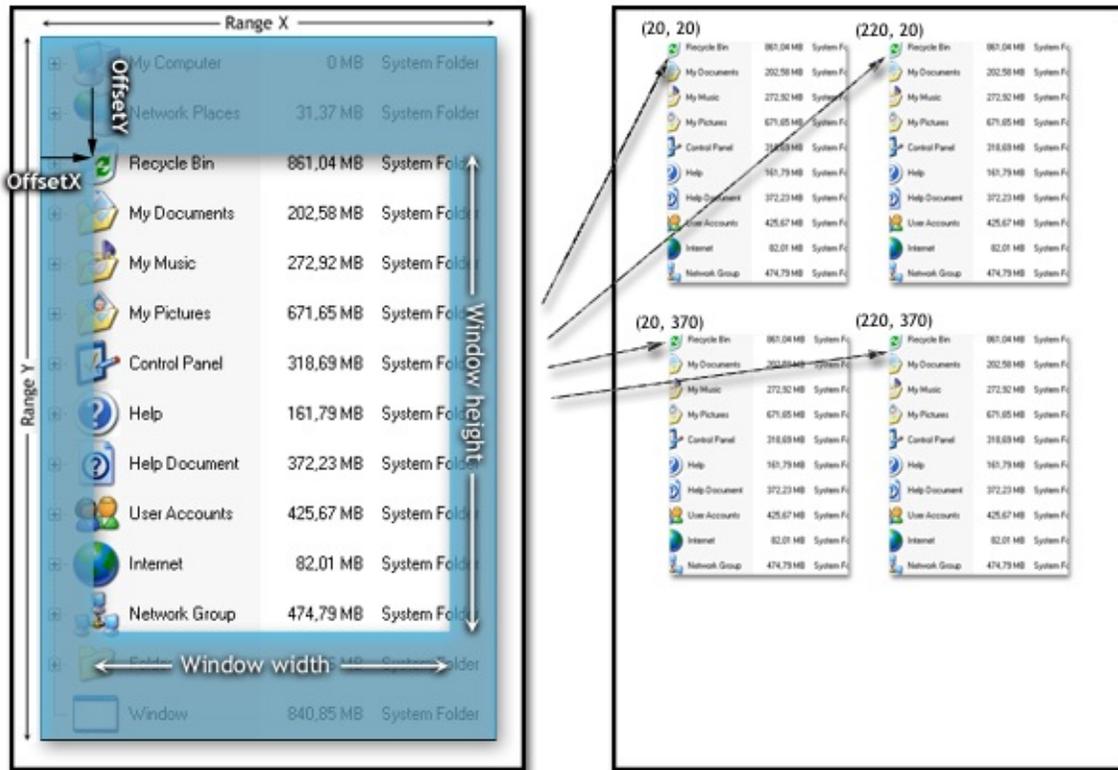
Some methods in Virtual Treeview work with an internal tree image, e.g. painting or hit determination. This tree image does not really exist but is rather an imagination of the entire tree drawn to an infinitely sized sheet. In this picture the tree is always drawn at position (0, 0) and advances to 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

Inner fundamentals

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

Inner fundamentals

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 underlying 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

cancellation.

Editing starts with the protected DoEdit method which may be triggered by the edit timer (which in turn is triggered by clicking again on the focused node), by pressing F2 or by calling EditNode. DoEdit creates an editor (actually only the edit link) via the virtual CreateEditor method which should be overridden by descendant trees to return a valid edit link (as [TVirtualStringTree](#) 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!](#)



Keyboard, hotkeys and incremental search

Inner fundamentals

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

Description

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

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

used to scroll the tree window.

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

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

		node does not change but the last visible column is focused.
	shift	Moves the focus to the last visible node (the focused column does not change) and includes every visible node, from the previously focused to the newly focused one, into the current selection. Modifications in grid mode: Not the focused node is changed but the last visible column is focused instead. The selection does not change.
	control	Scrolls the tree to the bottom right corner without change of selection or focused node.
Prior (page up)	none	Scrolls the tree window and single selects a node one page up. This node receive also the current focus.
	shift	Like without modifier key but includes a page of nodes into the current selection.
	control	Scrolls the tree window one page up without change of selection or focused node.
Next (page down)	none	Same as prior but one page down instead.
	shift	Same as prior but one page down instead.
	control	Same as prior but one page down instead.
Up	none	Advances the focus from the currently focused node to the previous visible node.
	shift	Advances the focus and adds the newly focused node to the current selection.
	control	Scrolls the tree window one line up. One line is defined as the DefaultNodeHeight.
Down	none	Same as up but one line down instead.

	shift	Same as up but one line down instead.
	control	Same as up but one line down instead.
Left	none	<p>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.</p> <p>Modifications in grid mode: If extended focus is enabled (see toExtendedFocus in Options) then the behavior changes to a simple navigation to the previous visible column.</p>
	shift	In opposition to the none-modifier case the expand state of the node does not matter nor is it changed. The focus is advanced in any case and sibling nodes as well as the parent node are added to the current selection.
	control	The tree window is scrolled to the left by the amount pixel given in the indent property.
Right	none	<p>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.</p> <p>Modifications in grid mode: If extended focus is enabled (see toExtendedFocus in Options) then the behavior changes to a simple navigation to the next visible column.</p>
		Same as the none-modifier case but the

	shift	selection is extended with the first child node.
	control	Same as left but the tree window is scrolled to the right.
Back	none	Moves the focus to the parent of the currently focused node and selects it as only node.
	shift	Modifier keys have no meaning for this case.
	control	Modifier keys have no meaning for this case.
Tabulator	none	The tabulator key is a bit special because it is only used with grid extensions to advance from cell to cell. Without modifier the focus changes from left to right and from top to bottom. It is necessary that you enable TAB support by setting property WantTabs to True.
	shift	Same as without modifier key but the focus advances backwards, from right to left and bottom to top.
	control	This modifier has no effect.
F1	none	This function key triggers node specific help support. Via the OnGetHelp event the application is queried for a help context to show.
	shift	This modifier has no effect.
	control	This modifier has no effect.
F2	none	This function key turns the tree view into edit mode if there is a focused node, the tree is editable and the application allows to edit the node.
	shift	This modifier has no effect.
	control	This modifier has no effect.

+ (add)	none	Expands the currently focused node.
	shift	This modifier alone has no effect, but see the following comment.
	control	Pressing the control key together with + will start auto sizing all columns in the tree. If the shift key is also pressed then the whole tree is expanded instead.
- (subtract)	none	Collapses the currently focused node.
	shift	This modifier alone has no effect, but see the following comment.
	control	Pressing the control key together with - will restore all columns to their previous widths. If the shift key is also pressed then the whole tree is collapsed instead.
* (multiply)	none	Expands the currently focused node and all its children and grand children.
	shift	This modifier has no effect.
	control	This modifier has no effect.
/ (divide)	none	Collapses the currently focused node and all its children and grand children.
	shift	This modifier has no effect.
	control	This modifier has no effect.
Escape	none	Stops actions which require a specific state in the tree like editing, mouse selecting, drag'n drop etc.
	shift	This modifier has no effect.
	control	This modifier has no effect.

Space	none	Used only if check support is enabled (see toCheckSupport in Options) and the currently focused node has got a check type other than ctNone. In this case the space key switches the check state.
	shift	This modifier has no effect.
	control	This modifier has no effect.
Apps (menu key)	none	Similar to F1 triggers the apps key popup menus on a node by node basis. For more information see also the event OnGetPopupMenu.
	shift	This modifier has no effect.
	control	This modifier has no effect.
A	none	This is the only "normal" character used as hotkey so far. It has only an effect together with the control key.
	shift	This modifier has no effect.
	control	Pressing 'A' together with the control key will select all currently visible nodes in the tree view.

Incremental search

Incremental search is a commonly used term to describe the effect that the user types some letters while the tree view has the focus and the control will try to locate a node whose caption matches the letters. Because Virtual Treeview does not know what caption a node has it cannot compare the incoming letters and uses therefore again an event to ask the application to do the comparison. By using the lesser of both string lengths and a partial comparison in this event the tree

will be able to select also partial matches. Note: Virtual Treeview tries to mimic the UI of the system list view and system tree view as close as possible and uses therefore two modes when searching. One is used when there is no key or only one key pressed and the new key is the same as the already recorded one. In this case the search always starts with the next node and only nodes which match the single new key will be found. This allows to quickly cycle through a number of nodes all matching/beginning with the same letter. The other mode is normal linear search where all key presses are recorded and compared with the nodes in the tree. Whenever the application considers a node as match (it even hasn't to have a caption the same as the search string) this node is returned as new target and focused.

Group

[Inner fundamentals](#)

Links

[Inner fundamentals](#)

What do you think about this topic? [Send feedback!](#)



Drag'n drop and clipboard handling

Inner fundamentals

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

Inner fundamentals

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 OnBeforeItemPaint 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!](#)



Inner fundamentals | A little code
repository by John Knipper

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

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.

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

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

1. Typ is an integer value, from which I can determine what kind of node this is, in my example 1, 2 or 3.
2. Name is the name of the node. This will be needed relatively often because it is also seen as part of the tree and I want to access this

information easily (man, I am lazy).

3. The pointer allows (similar to the data property of the tree) a record or even better a class instance to connect.

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

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

Alternative solution:

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

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

So I do following:

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

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

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

type

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

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

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

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

Implementation

An empty tree

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

- Either an existing tree is read from a file or

- A top-level node is created.

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

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

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

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

```
begin
  // load tree
  ....
end;
....
end;
```

Data for the node

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

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

begin
  ...
  // determine data for node
  NodeData := BookmarkForm.BookmarkTree.GetNodeData
  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 determine the index to determine which image from 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 an image list to the property StateImages) the value for this case (Kind \diamond ikState) is -1.

```

procedure TBookmarkForm.BookmarkTreeGetImageIndex(Sender: TVirtualNode;
  Node: PVirtualNode; Kind: TVTImageKind; Column: Integer)
var
  NodeData: ^TTreeData;
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 node
    pNodeData: Pointer;
end;
```

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

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

and then a structure of the form:

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

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

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

```
if NodeD.BasicND = nil then  
    Text := ''
```

must be returned or you wrap the entire initialization into a `BeginUpdate/EndUpdate` block and initialized the nodes

before EndUpdate is called (e.g. by ValidateNode(Node)).*

Without this provision an access violation would be the result.

Example class declaration

```
unit TreeData;

interface

//=====

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

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

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

```

    end;

implementation

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

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

end.

```

Example creation of the tree

```

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

var
    Node: PVirtualNode;
    NodeD: ^rTreeData;

begin
    ....
    // create tree
    MyTree.NodeDataSize := SizeOf(rTreeData);

```

```

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

// returns the text (the identification) of the node
procedure TMyForm.MyTreeGetText(Sender: TBaseVirtualNode;
    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: TBaseVirtualNode;
    Node: PVirtualNode; Kind: TVTImageKind; Column: Integer;
    var Index: Integer);
var

```

```

NodeD: ^rTreeData;

begin
  NodeD := Sender.GetNodeData(Node);

  case Kind of
    ikState: // for the case the state index has be
      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 = Send

```

Implementation of the method:

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

```
constructor TBasicNodeData.Create(vName: ShortString)
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 it shall be possible to add and remove nodes.

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

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

```
procedure TMyForm.addClick (Sender: TObject);  
  
var  
    Node: PVirtualNode;  
    NodeD: ^rTreeData;  
  
begin  
    // Ok, a node must be added.  
    Node := MyTree.AddChild(MyTree.FocusedNode); // 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 BookmarkTree.FocusedNode = nil then
begin
    // insert as child of the first top level node
    Node := BookmarkTree.AddChild(BookmarkTree.Root)
    // determine data for node
    NodeD := BookmarkTree.GetNodeData(Node);
    NodeD.BasicND := TFolderNodeData.Create('new fo
end
else
begin
    // Ok, a new node must be added.
    Node := BookmarkTree.AddChild(BookmarkTree.Focu
    // determine data of the node
    NodeD := BookmarkTree.GetNodeData(Node);
    NodeD.BasicND := TFolderNodeData.Create('new fo
end;

```

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

```

procedure TMyForm.delClick (Sender: TObject);
begin
    // The focused node should be removed. The top le
    if MyTree.FocusedNode = nil then
        MessageDlg('There was no node selected.', mtInf
    else
        // Note: RootNode is the internal (hidden) root

```

```

// level nodes. To determine whether a node is
// GetNodeLevel which returns 0 for top level n
if MyTree.FocusedNode.Parent = MyTree.RootNode th
    MessageDlg('The project node must not be delete
else
    MyTree.DeleteNode(MyTree.FocusedNode);
end;

```

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

While we are at deleting nodes:

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

```

procedure TMyForm.MyTreeFreeNode(Sender: TBaseVirtu
begin
    // Free here the node data (Note: type PtreeData

```

```
PTreeData(Sender.GetNodeData(Node)).BasicND.Free;  
end;
```

Adding folder and leafs

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

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

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

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

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

Storing the tree

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

Again a quote from Preparations:

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

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

Hint:

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

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

For saving the tree the procedure

```
procedure TBaseVirtualTree.SaveToFile(const FileName
```

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

```
property OnSaveNode: TVTSaveNodeEvent read FOnSaveN
```

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

```
procedure TMyForm.MyTreeSaveNode(Sender: TBaseVirtu  
begin
```

```
PTreeData(Sender.GetNodeData(Node)).BasicND.SaveT  
end;
```

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

```
procedure TBasicNodeData.SaveNode(Stream: TStream);  
  
var  
    size: Integer;  
  
begin  
    // save type of the node  
    Stream.Write(Art, SizeOf(Art));  
  
    // store cName  
    Size := Length(cName) + 1; // include terminating  
    Stream.Write(Size, SizeOf(Size)); // store length  
    Stream.Write(PChar(cName)^, Size); // now the str  
  
    // store cImageIndex  
    Stream.Write(cImageIndex, SizeOf(cImageIndex));  
  
    // store cImageIndexFocus  
    Stream.Write(cImageIndexFocus, SizeOf(cImageIndex  
  
    // store cChildrenAllowed  
    Stream.Write(cChildrenAllowed, SizeOf(cChildrenAL  
end;
```

Now we can the tree we save also load again. This p

```
try
```

```

// load tree
MyTree.LoadFromFile(MainControlForm.FileName);
except
  on E: Exception do
    begin
      Application.MessageBox(PChar(E.Message), PChar(
MainControlForm.FileName := '');

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

```

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

```

procedure TBasicNodeData.LoadNode(Stream: TStream);

var
  Size: Integer;
  StrBuffer: PChar;

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

  StrBuffer := AllocMem(Size); // get temporary mem
  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(cImageIndexF  
  
// load cChildrenAllowed  
Stream.Read(cChildrenAllowed, SizeOf(cChildrenAll  
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: TBaseVirtual
  Column: Integer; TextType: TVSTTextType; var Text

var
  NodeD: ^rTreeData;

begin
  NodeD := Sender.GetNodeData(Node);

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

procedure TMyForm.MyTreeGetImageIndex(Sender: TBase
  Kind: TVTImageKind; Column: Integer; var Index: I

var
  NodeD: ^rTreeData;

begin
  NodeD := Sender.GetNodeData(Node);

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

```

```

        ikNormal, ikSelected:
            Index := NodeD.BasicND.GetImageIndex(Node =
ikOverlay: // e.g. to mark a node whose conte
            // Note: don't forget to call ImageList.Ove
            if NodeD.BasicND.ImageIndex = 4 then
                Index := 6;
        end;
end;

```

Accessing the columns

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

```

procedure TBookmarkForm.BookmarkTreeColumnResize(Sender: TTreeView)
var
    NodeD: PTreeData;
begin
    NodeD := Sender.GetNodeData(Sender.RootNode.FirstChild);
    // Keep the new size of the column in the project
    TProjectNodeData(NodeD.BasicND).SetHColumnsWidth(
        TVirtualStringTree(Sender).Header.Columns.Items[0].Width);
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:

```
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:

```
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:

```
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: TBaseVirtual  
  
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  
                else Level := 0;
```

```

case FFillMode of
  0: // fill tree with a specific amount of nodes
    begin
      // determine new node level
      if Level < (LevelsUpDown.Position - 1) then
        end;
  1: // fill tree with one million root nodes (
    ;
  2: // fill tree with a certain amount of root nodes
    begin
      Data.FixedText := True;
      Data.NewText := Format('Node: %d', [Node.Level]);
    end;
  3: // fill tree with a certain amount of root nodes
      // up to an absolute amount of ~1 million nodes
    begin
      if Assigned(ParentNode) then Count := ParentNode.Count
      else Count := TVi
    end;
      if (Level < 15) and
        (Random(Count) < (Count div 2)) and
        (FCurrentCount < 1000000) then Include(ParentNode);
    end;
end;

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

```

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

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

E: Either use

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

Or in that case you can use

```
var
  NodeData: PTreeData;

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

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

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

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

```
procedure TFRM_WWW_main.vFavTreeGetText(Sender: TBase  
  
begin  
    // Column is -1 if the header is hidden or no col  
    if Column < 0 then Exit;  
    if TVirtualStringTree(Sender).Header.Columns[Colu  
    begin  
        Table.GotoBookmark(TBookmark(Sender.GetNodeData  
        Text := Table.FieldName('Name').asString;  
    end;  
end;
```

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

A:

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

Q: How can I display text for other columns?

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

E:

```
procedure TFRM_WWW_main.vFavTreeGetText(Sender: TBase
```

```

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: TObject; Item: TTreeNode; var Index: Integer; Kind: TImageKind);
begin
  if Kind = ikState then
  begin
    Index := 2;
  end
  else
  begin
    if (Kind = ikNormal) or (Kind = ikSelected) then
    begin
      Index := 1;
    end;
  end;
end;

```

or just use

```
procedure TFRM_WWW_main.vFavTreeGetImageIndex(Sende
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!](#)



A little code repository by John
Knipper | Licensing

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!](#)



Classes

[Classes](#) | [Virtual Treeview](#) | [Legend](#)

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

Classes

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

Description

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

Group

Classes

Class Hierarchy



File

VirtualTrees

Links

Classes

What do you think about this topic? [Send feedback!](#)



TBaseVirtualTree Class

[Events](#) | [Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

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

Pascal

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

Description

This class implements most of the base features and abilities and can be used to derive new classes, which want to hide most of the details of the tree, which other descendants like [TVirtualStringTree](#) 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

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.

➤ **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.

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 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**
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.
-  **ColumnsIsEmpty**
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.
- 🔴🔵🟢 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.

  **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 s 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.

 **HandleMouseDown**

Not documented.

 **HandleMouseUp**

Not documented.

 **HasAsParent**

Not documented.

 **HasImage**

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

 **HasPopupMenu**

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

```
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 Class](#) |
[TBaseVirtualTree.Alignment](#)
Property |
[TBaseVirtualTree.AutoExpandDelay](#)
Property

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.AnimationDuration](#)
[Property](#) |
[TBaseVirtualTree.AutoScrollDelay](#)
[Property](#)

TBaseVirtualTree.AutoExpandDelay Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.AutoExpandDelay](#)
[Property](#) |
[TBaseVirtualTree.AutoScrollInterval](#)
[Property](#)

TBaseVirtualTree.AutoScrollDelay Property

[TBaseVirtualTree Class](#)

Time which determines when auto scrolling should start.

Pascal

```
property AutoScrollDelay: Cardinal;
```

Description

Once the mouse pointer has been moved near to a border a timer is started using the interval specified by `AutoScrollDelay`. When the timer has fired auto scrolling starts provided it is enabled (see also [TreeOptions](#)). The value is specified in milliseconds.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.AutoScrollDelay](#)
Property |
[TBaseVirtualTree.Background](#)
Property

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 Class](#) |
[TBaseVirtualTree.AutoScrollInterval](#)
[Property](#) |
[TBaseVirtualTree.BackgroundOffsetX](#)
[Property](#)

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 Class](#) |
[TBaseVirtualTree.Background](#)
[Property](#) |
[TBaseVirtualTree.BackgroundColorOffsetY](#)
[Property](#)

TBaseVirtualTree.BackgroundColorOffsetX Property

[TBaseVirtualTree Class](#)

Horizontal offset of the background image.

Pascal

```
property BackgroundOffsetX: Integer;
```

Description

Determines the horizontal offset of the left border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.BackgroundColorX](#)
[Property](#) |
[TBaseVirtualTree.BorderStyle](#)
[Property](#)

TBaseVirtualTree.BackgroundColorY Property

[TBaseVirtualTree Class](#)

Vertical offset of the background image.

Pascal

```
property BackgroundOffsetY: Integer;
```

Description

Determines the vertical offset of the top border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.BackgroundColorY](#)
Property |
[TBaseVirtualTree.ButtonFillMode](#)
Property

TBaseVirtualTree.BorderStyle Property

[TBaseVirtualTree Class](#)

Same as TForm.BorderStyle.

Pascal

```
property BorderStyle: TBorderStyle;
```

Description

See TForm.BorderStyle.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.BorderStyle](#)
[Property](#) |
[TBaseVirtualTree.ButtonStyle](#)
[Property](#)

TBaseVirtualTree.ButtonFillMode Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ButtonFillMode](#)
[Property](#) |
[TBaseVirtualTree.ChangeDelay](#)
[Property](#)

TBaseVirtualTree.ButtonStyle Property

[TBaseVirtualTree Class](#)

Determines the look of node buttons.

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 Class](#) |
[TBaseVirtualTree.ButtonStyle](#)
[Property](#) |
[TBaseVirtualTree.CheckImageKind](#)
[Property](#)

TBaseVirtualTree.ChangeDelay Property

[TBaseVirtualTree Class](#)

Time which determines when the [OnChange](#) event should be triggered after the actual change event.

Pascal

```
property ChangeDelay: Cardinal;
```

Description

In order to accumulate many quick changes in the tree you can use this delay value to specify after which wait time the [OnChange](#) event should occur. A value of 0 means to trigger [OnChange](#) immediately after the change (usually a selection or focus change) 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ChangeDelay](#)
Property |
[TBaseVirtualTree.CheckImages](#)
Property

TBaseVirtualTree.CheckImageKind Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.CheckImageKind](#)
[Property](#) |
[TBaseVirtualTree.CheckState](#)
[Property](#)

TBaseVirtualTree.CheckImages Property

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
property CheckImages: TCustomImageList;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.CheckImages](#)
[Property](#) |
[TBaseVirtualTree.CheckType](#)
[Property](#)

TBaseVirtualTree.CheckState Property

[TBaseVirtualTree Class](#)

Read or set the check state of a node.

Pascal

```
property CheckState [Node: PVirtualNode]: TCheckStat
```

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 Class](#) |
[TBaseVirtualTree.CheckState](#)
Property |
[TBaseVirtualTree.ChildCount](#)
Property

TBaseVirtualTree.CheckType Property

[TBaseVirtualTree Class](#)

Read or set the check type of a node.

Pascal

```
property CheckType [Node: PVirtualNode]: TCheckType;
```

Description

The CheckType property can be used to read the current check type of a node or to set a new one. Setting a new check type will reset a the node's check state to csUncheckedNormal.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.CheckType](#)
[Property](#) |
[TBaseVirtualTree.ChildrenInitialized](#)
[Property](#)

TBaseVirtualTree.ChildCount Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property ChildCount [Node: PVirtualNode]: Cardinal;
```

Description

ChildCount can be used to read the current number of child nodes or to change it. Assigning a lower value than there was before will automatically delete as many child nodes (starting from the last child) as there are more than what was set. Increasing the value will add new child nodes. Note: code behind this property is very effective, so it using ChildCount is highly recommended over manipulating the child count using [AddChild](#), [InsertNode](#) and [DeleteNode](#).

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ChildCount](#)
[Property](#) |
[TBaseVirtualTree.ClipboardFormats](#)
[Property](#)

TBaseVirtualTree.ChildrenInitialized Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property ChildrenInitialized [Node: PVirtualNode]: B
```

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 Class](#) |
[TBaseVirtualTree.ChildrenInitialized
Property](#) | [TBaseVirtualTree.Colors
Property](#)

TBaseVirtualTree.ClipboardFormats Property

[TBaseVirtualTree Class](#)

Special class to keep a list of clipboard format descriptions.

Pascal

```
property ClipboardFormats: TClipboardFormats;
```

Description

This TStringList descendant is used to keep a number of clipboard format descriptions, which are usually used to register clipboard formats with the system. Using a string list for this task allows to store enabled clipboard formats in the DFM.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ClipboardFormats](#)
[Property](#) |
[TBaseVirtualTree.CustomCheckImages](#)
[Property](#)

TBaseVirtualTree.Colors Property

[TBaseVirtualTree Class](#)

A collection of colors used in the tree.

Pascal

```
property Colors: TVTColors;
```

Description

This property holds an instance of the [TVTColors](#) class, which is used to customize many of the colors used in a tree. Placing them all in a specialized class helps organizing the colors in the object inspector and improves general management.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.Colors Property](#) |
[TBaseVirtualTree.DefaultNodeHeight Property](#)

TBaseVirtualTree.CustomCheckImages Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.CustomCheckImages](#)
[Property](#) |
[TBaseVirtualTree.DefaultPasteMode](#)
[Property](#)

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 Class](#) |
[TBaseVirtualTree.DefaultNodeHeight](#)
[Property](#) |
[TBaseVirtualTree.DragHeight](#)
[Property](#)

TBaseVirtualTree.DefaultPasteMode Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DefaultPasteMode](#)
[Property](#) |
[TBaseVirtualTree.DragImage](#)
[Property](#)

TBaseVirtualTree.DragHeight Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DragHeight](#)
[Property](#) |
[TBaseVirtualTree.DragImageKind](#)
[Property](#)

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 Class](#) |
[TBaseVirtualTree.DragImage](#)
[Property](#) |
[TBaseVirtualTree.DragManager](#)
[Property](#)

TBaseVirtualTree.DragImageKind Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property DragImageKind: TVTDragImageKind;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DragImageKind](#)
[Property](#) |
[TBaseVirtualTree.DragOperations](#)
[Property](#)

TBaseVirtualTree.DragManager Property

[TBaseVirtualTree Class](#) | [See Also](#)

Holds the reference to the internal drag manager.

Pascal

```
property DragManager: IVTDragManager;
```

Description

The drag manager is the central point for the drag'n drop support in Virtual Treeview. Usually you do not need to access it but sometimes it might be necessary so the reference is accessible through this property.

See Also

[TVTDragManager](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DragManager](#)
[Property](#) |
[TBaseVirtualTree.DragSelection](#)
[Property](#)

TBaseVirtualTree.DragOperations Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DragOperations](#)
[Property](#) |
[TBaseVirtualTree.DragType](#)
[Property](#)

TBaseVirtualTree.DragSelection Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property DragSelection: TNodeArray;
```

Description

This list is a local copy of the current selection array and is only used during a drag operation.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DragSelection](#)
[Property](#) |
[TBaseVirtualTree.DragWidth](#)
[Property](#)

TBaseVirtualTree.DragType Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property DragType: TVTDragType;
```

Description

Traditionally, Delphi only supports its own drag mechanism, which is not compatible with the rest of the system. This VCL **dragging** also does not support to transport random data nor does it support drag operations between applications. Thus Virtual Treeview also supports the generally used OLE **dragging**, which in turn is incompatible with VCL **dragging**. Depending on your needs you can enable either VCL or OLE **dragging** as both together cannot be started. However, Virtual Treeview is able to act as drop target for both kind of data, independant of what is set in DragType.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DragType Property](#) |
[TBaseVirtualTree.DrawSelectionMode Property](#)

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 Class](#) |
[TBaseVirtualTree.DragWidth](#)
[Property](#) |
[TBaseVirtualTree.DropTargetNode](#)
[Property](#)

TBaseVirtualTree.DrawSelectionMode Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property DrawSelectionMode: TVTDrawSelectionMode;
```

Description

Virtual 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 Class](#) |
[TBaseVirtualTree.DrawSelectionMode](#)
[Property](#) |
[TBaseVirtualTree.EditColumn](#)
[Property](#)

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 Class](#) |
[TBaseVirtualTree.DropTargetNode](#)
[Property](#) |
[TBaseVirtualTree.EditDelay](#)
[Property](#)

TBaseVirtualTree.EditColumn Property

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
property EditColumn: TColumnIndex;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.EditColumn](#)
[Property](#) |
[TBaseVirtualTree.EditLink](#)
[Property](#)

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

```
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 Class](#) |
[TBaseVirtualTree.EditDelay](#)
[Property](#) |
[TBaseVirtualTree.Expanded](#)
[Property](#)

TBaseVirtualTree.EditLink Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property EditLink: IVTEditLink;
```

Description

During an edit operation a link is established between the tree and the editor for the current node. By default a simple TEdit control is used as editor but due to the great customization possibilities there can be any node editor you may want. In order to communicate with this potentially unknown node editor the edit link is used. The EditLink property holds this link during the edit operation, so you can manipulate the interface.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.EditLink](#)
[Property](#) |
[TBaseVirtualTree.FocusedColumn](#)
[Property](#)

TBaseVirtualTree.Expanded Property

[TBaseVirtualTree Class](#)

Read or set the expanded state of a particular node.

Pascal

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

Description

Using this property you can expand or collapse the given node. This method uses the central [ToggleNode](#) method.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.Expanded](#)
[Property](#) |
[TBaseVirtualTree.FocusedNode](#)
[Property](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.FocusedColumn
Property](#) | [TBaseVirtualTree.Font
Property](#)

TBaseVirtualTree.FocusedNode Property

[TBaseVirtualTree Class](#) | [See Also](#)

Read or set the currently focused node.

Pascal

```
property FocusedNode: PVirtualNode;
```

Description

One node (and only one) in the tree view can have the current input focus, marked as dotted rectangle around the node's caption. Having the input focus means this node can be edited by pressing F2 or clicking on it and user keyboard input is interpreted with respect to the focused node (e.g. tree navigation, expansion/collapsing etc.). If extended focus is enabled then also the [FocusedColumn](#) 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 Class](#) |
[TBaseVirtualTree.FocusedNode](#)
[Property](#) |
[TBaseVirtualTree.FullyVisible](#)
[Property](#)

TBaseVirtualTree.Font Property

[TBaseVirtualTree Class](#)

Same as TWinControl.Font.

Pascal

```
property Font;
```

Description

See TWinControl.Font.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.Font Property](#) |
[TBaseVirtualTree.HasChildren Property](#)

TBaseVirtualTree.FullyVisible Property

[TBaseVirtualTree Class](#) | [See Also](#)

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

Pascal

```
property FullyVisible [Node: PVirtualNode]: Boolean;
```

Description

Beside the fact that a node can be out of the client area there are two possibilities for it to be hidden. One is the `vsVisible` state in [TVirtualNodeState](#), 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 Class](#) |
[TBaseVirtualTree.FullyVisible](#)
[Property](#) |
[TBaseVirtualTree.Header Property](#)

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 navigating 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 Class](#) |
[TBaseVirtualTree.HasChildren](#)
[Property](#) |
[TBaseVirtualTree.HeaderRect](#)
[Property](#)

TBaseVirtualTree.Header Property

[TBaseVirtualTree Class](#) | [See Also](#)

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 Class](#) |
[TBaseVirtualTree.Header Property](#)
| [TBaseVirtualTree.HintAnimation
Property](#)

TBaseVirtualTree.HeaderRect Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property HeaderRect: TRect;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.HeaderRect](#)
[Property](#) |
[TBaseVirtualTree.HintMode](#)
[Property](#)

TBaseVirtualTree.HintAnimation Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.HintAnimation](#)
[Property](#) |
[TBaseVirtualTree.HotCursor](#)
[Property](#)

TBaseVirtualTree.HintMode Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.HintMode](#)
[Property](#) |
[TBaseVirtualTree.HotNode](#)
[Property](#)

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 Class](#) |
[TBaseVirtualTree.HotCursor](#)
[Property](#) |
[TBaseVirtualTree.Images Property](#)

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 Class](#) |
[TBaseVirtualTree.HotNode Property](#) |
[TBaseVirtualTree.IncrementalSearch Property](#)

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!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.Images Property](#) |
[TBaseVirtualTree.IncrementalSearchDirection](#)
[Property](#)

TBaseVirtualTree.IncrementalSearch Property

[TBaseVirtualTree Class](#) | [See Also](#)

Read or set the current incremental search mode.

Pascal

```
property IncrementalSearch: TVTIncrementalSearch;
```

Description

Virtual Treeview can do an incremental search by calling back the application when comparing node captions. The `IncrementalSearch` property determines whether incremental search is enabled and which nodes should be searched through.

See Also

[IncrementalSearchDirection](#), [IncrementalSearchStart](#),
[IncrementalSearchTimeout](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.IncrementalSearch](#)
[Property](#) |
[TBaseVirtualTree.IncrementalSearchStart](#)
[Property](#)

TBaseVirtualTree.IncrementalSearchDirection Property

[TBaseVirtualTree Class](#) | [See Also](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.IncrementalSearchDirection](#)
[Property](#) |
[TBaseVirtualTree.IncrementalSearchTimeout](#)
[Property](#)

TBaseVirtualTree.IncrementalSearchStart Property

[TBaseVirtualTree Class](#) | [See Also](#)

Read or set where to start incremental search.

Pascal

```
property IncrementalSearchStart: TVTSearchStart;
```

Description

When incremental search is enabled in the tree view then you can specify here, where to start the next incremental search operation from.

See Also

[IncrementalSearch](#), [IncrementalSearchDirection](#),
[IncrementalSearchTimeout](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.IncrementalSearchStart
Property](#) | [TBaseVirtualTree.Indent
Property](#)

TBaseVirtualTree.IncrementalSearchTime Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

TBaseVirtualTree Class, See Also

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.IncrementalSearchTimeout
Property](#) | [TBaseVirtualTree.IsDisabled
Property](#)

TBaseVirtualTree.Indent Property

[TBaseVirtualTree Class](#)

Read or set the indentation amount for node levels.

Pascal

```
property Indent: Cardinal;
```

Description

Each new level in the tree (child nodes of a parent node) are visually shifted to distinguish 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 Class](#) |
[TBaseVirtualTree.Indent Property](#) |
[TBaseVirtualTree.IsVisible Property](#)

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 Class](#) |
[TBaseVirtualTree.IsDisabled](#)
[Property](#) |
[TBaseVirtualTree.LastClickPos](#)
[Property](#)

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 Class](#) |
[TBaseVirtualTree.IsVisible](#)
[Property](#) |
[TBaseVirtualTree.LastDropMode](#)
[Property](#)

TBaseVirtualTree.LastClickPos Property

[TBaseVirtualTree Class](#)

Used for retained drag start and wheel mouse scrolling.

Pascal

```
property LastClickPos: TPoint;
```

Description

This internal positions is made public to allow descendants to modify mainly the right click behavior of the tree control.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.LastClickPos](#)
Property |
[TBaseVirtualTree.LineMode](#)
Property

TBaseVirtualTree.LastDropMode Property

[TBaseVirtualTree Class](#)

Read how the last drop operation finished.

Pascal

```
property LastDropMode: TDropMode;
```

Description

In the case you don't handle drag'n drop operations directly in [OnDragDrop](#) 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 Class](#) |
[TBaseVirtualTree.LastDropMode](#)
[Property](#) |
[TBaseVirtualTree.LineStyle](#)
[Property](#)

TBaseVirtualTree.LineMode Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.LineMode](#)
[Property](#) |
[TBaseVirtualTree.Margin Property](#)

TBaseVirtualTree.LineStyle Property

[TBaseVirtualTree Class](#)

Read or set the mode of the tree lines.

Pascal

```
property LineStyle: TVTLineStyle;
```

Description

Virtual Treeview allows to customize the lines used to display the node hierarchy. The default style is a dotted pattern, but you can also make solid lines or specify your own line pattern.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.LineStyle](#)
[Property](#) |
[TBaseVirtualTree.MultiLine](#)
[Property](#)

TBaseVirtualTree.Margin Property

[TBaseVirtualTree Class](#) | [See Also](#)

Read or set the tree's node margin.

Pascal

```
property Margin: Integer;
```

Description

The node margin is the distance between the cell bounds and its content like the lines, images, check box and so on. However this border is only applied to the left and right side of the node cell.

Note: there is also a [TextMargin](#) property in [TVirtualStringTree](#), which is an additional border for the cell text only.

See Also

[TVirtualStringTree.TextMargin](#)

Class

[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

```
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 Class](#) |
[TBaseVirtualTree.MultiLine](#)
[Property](#) |
[TBaseVirtualTree.NodeDataSize](#)
[Property](#)

TBaseVirtualTree.NodeAlignment Property

[TBaseVirtualTree Class](#) | [See Also](#)

Read or set the node alignment value.

Pascal

```
property NodeAlignment: TVTNodeAlignment;
```

Description

Nodes have got an align member, which is used to determine the vertical position of the node's images and tree lines. The NodeAlignment property specifies how to interpret the value in the align member.

See Also

[TVirtualNode](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.NodeAlignment](#)
Property |
[TBaseVirtualTree.NodeHeight](#)
Property

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.NodeDataSize](#)
[Property](#) |
[TBaseVirtualTree.NodeParent](#)
[Property](#)

TBaseVirtualTree.NodeHeight Property

[TBaseVirtualTree Class](#)

Read or set a node's height.

Pascal

```
property NodeHeight [Node: PVirtualNode]: Cardinal;
```

Description

Each node can have its individual height, which is stored in the node's record. You could directly assign a value to this member but I strongly discourage this as it does not update certain other structures in the tree. Instead use the NodeHeight property here to modify a node's height.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.NodeHeight](#)
[Property](#) |
[TBaseVirtualTree.OffsetXY](#)
[Property](#)

TBaseVirtualTree.NodeParent Property

[TBaseVirtualTree Class](#) | [See Also](#)

Read or set a node's parent node.

Pascal

```
property NodeParent [Node: PVirtualNode]: PVirtualNo
```

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 Class](#) |
[TBaseVirtualTree.NodeParent Property](#) |
[TBaseVirtualTree.OnAdvancedHeaderDraw](#)
[Event](#)

TBaseVirtualTree.OffsetXY Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property OffsetX: Integer;  
property OffsetXY: TPoint;  
property OffsetY: Integer;
```

Description

Virtual Treeview allows to retrieve or set the internal scroll offset directly, without sending WM_HSCROLL/WM_VSCROLL message around. This allows also to link two or more trees together. This scroll offset is given in pixels and is always less or equal 0.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OffsetXY](#)
[Property](#) |
[TBaseVirtualTree.OnAfterCellPaint](#)
[Event](#)

TBaseVirtualTree.OnAdvancedHeaderDraw Event

[TBaseVirtualTree Class](#) | [See Also](#)

Header paint support event.

Pascal

```
property OnAdvancedHeaderDraw: TVTAdvancedHeaderPaint;
```

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 Class](#) |
[TBaseVirtualTree.OnAdvancedHeaderDraw
Event](#) | [TBaseVirtualTree.OnAfterItemErase
Event](#)

TBaseVirtualTree.OnAfterCellPaint Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnAfterCellPaint](#)
[Event](#) |
[TBaseVirtualTree.OnAfterItemPaint](#)
[Event](#)

TBaseVirtualTree.OnAfterItemErase Event

[TBaseVirtualTree Class](#) | [See Also](#)

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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnAfterItemErase](#)
[Event](#) |
[TBaseVirtualTree.OnAfterPaint](#)
[Event](#)

TBaseVirtualTree.OnAfterItemPaint Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnAfterItemPaint](#)
[Event](#) |
[TBaseVirtualTree.OnBeforeCellPaint](#)
[Event](#)

TBaseVirtualTree.OnAfterPaint Event

[TBaseVirtualTree Class](#) | [See Also](#)

Paint support event.

Pascal

```
property OnAfterPaint: TVTPaintEvent;
```

Description

Called after all nodes which needed an update have been drawn. This event is called once per paint cycle.

See Also

[Paint cycles and stages](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnAfterPaint Event](#) |
[TBaseVirtualTree.OnBeforeItemErase Event](#)

TBaseVirtualTree.OnBeforeCellPaint Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnBeforeCellPaint](#)
[Event](#) |
[TBaseVirtualTree.OnBeforeItemPaint](#)
[Event](#)

TBaseVirtualTree.OnBeforeItemErase Event

[TBaseVirtualTree Class](#) | [See Also](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnBeforeItemErase](#)
[Event](#) |
[TBaseVirtualTree.OnBeforePaint](#)
[Event](#)

TBaseVirtualTree.OnBeforeItemPaint Event

[TBaseVirtualTree Class](#) | [See Also](#)

Paint support event.

Pascal

```
property OnBeforeItemPaint: TVTBeforeItemPaintEvent;
```

Description

Called after the background of a node has been drawn and just before the node itself is painted. In this event the application gets the opportunity to decide whether a node should be drawn normally or should be skipped. The application can draw the node itself if necessary or leave the node area blank.

See Also

[Paint cycles and stages](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnBeforeItemPaint
Event](#) | [TBaseVirtualTree.OnChange
Event](#)

TBaseVirtualTree.OnBeforePaint Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnBeforePaint](#)
[Event](#) |
[TBaseVirtualTree.OnChecked](#)
[Event](#)

TBaseVirtualTree.OnChange Event

[TBaseVirtualTree Class](#)

Navigation support event.

Pascal

```
property OnChange: 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 Class](#) |
[TBaseVirtualTree.OnChange](#)
[Event](#) |
[TBaseVirtualTree.OnChecking](#)
[Event](#)

TBaseVirtualTree.OnChecked Event

[TBaseVirtualTree Class](#)

Check support event.

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 Class](#) |
[TBaseVirtualTree.OnChecked](#)
[Event](#) |
[TBaseVirtualTree.OnCollapsed](#)
[Event](#)

TBaseVirtualTree.OnChecking Event

[TBaseVirtualTree Class](#)

Check support event.

Pascal

```
property OnChecking: TVTCheckChangingEvent;
```

Description

Triggered when a node's check state is about to change and allows to prevent the change.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnChecking](#)
[Event](#) |
[TBaseVirtualTree.OnCollapsing](#)
[Event](#)

TBaseVirtualTree.OnCollapsed Event

[TBaseVirtualTree Class](#)

Miscellaneous event.

Pascal

```
property OnCollapsed: TVTChangeEvent;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnCollapsed](#)
[Event](#) |
[TBaseVirtualTree.OnColumnClick](#)
[Event](#)

TBaseVirtualTree.OnCollapsing Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnCollapsing](#)
[Event](#) |
[TBaseVirtualTree.OnColumnDbClick](#)
[Event](#)

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 Class](#) |
[TBaseVirtualTree.OnColumnClick](#)
[Event](#) |
[TBaseVirtualTree.OnColumnResize](#)
[Event](#)

TBaseVirtualTree.OnColumnDbClick Event

[TBaseVirtualTree Class](#) | [See Also](#)

Header and column support event.

Pascal

```
property OnColumnDbClick: TVTColumnDbClickEvent;
```

Description

Same as [OnColumnClick](#) but for double clicks.

See Also

[OnColumnClick](#), [OnHeaderDbClick](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnColumnDbClick](#)
[Event](#) |
[TBaseVirtualTree.OnCompareNodes](#)
[Event](#)

TBaseVirtualTree.OnColumnResize Event

[TBaseVirtualTree Class](#)

Header and column support routine.

Pascal

```
property OnColumnResize: TVTHeaderNotifyEvent;
```

Description

Triggered when a column is being resized. During resize OnColumnResize is frequently hence you should make any code in the associated event handle a short and fast as possible.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnColumnResize](#)
[Event](#) |
[TBaseVirtualTree.OnCreateDataObject](#)
[Event](#)

TBaseVirtualTree.OnCompareNodes Event

[TBaseVirtualTree Class](#) | [See Also](#)

Sort and search support event.

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: TBaseV
  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 Class](#) |
[TBaseVirtualTree.OnCompareNodes](#)
[Event](#) |
[TBaseVirtualTree.OnCreateDragManager](#)
[Event](#)

TBaseVirtualTree.OnCreateDataObject Event

[TBaseVirtualTree Class](#)

Drag'n drop support event.

Pascal

```
property OnCreateDataObject: TVTCreateDataObjectEvent
```

Description

This event is called when the tree's drag manager needs a data object interface to start a drag'n drop operation. 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 Class](#) |
[TBaseVirtualTree.OnCreateDataObject](#)
[Event](#) |
[TBaseVirtualTree.OnCreateEditor](#)
[Event](#)

TBaseVirtualTree.OnCreateDragManager Event

[TBaseVirtualTree Class](#)

Drag'n drop support event.

Pascal

```
property OnCreateDragManager: TVTCreateDragManagerEv
```

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 Class](#) |
[TBaseVirtualTree.OnCreateDragManager
Event](#) | [TBaseVirtualTree.OnDragAllowed
Event](#)

TBaseVirtualTree.OnCreateEditor Event

[TBaseVirtualTree Class](#) | [See Also](#)

Editing support event.

Pascal

```
property OnCreateEditor: TVTCreateEditorEvent;
```

Description

Allows to supply a customized node editor without changing the tree. [TBaseVirtualTree](#) triggers this event and raises an exception if there no editor is returned. If you don't want this then disable edit support for nodes in `TreeOptions.MiscOptions`. Descendants like [TCustomVirtualStringTree](#) supply a generic and simple string editor.

See Also

[Editors and editing](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnCreateEditor](#)
[Event](#) |
[TBaseVirtualTree.OnDragDrop](#)
[Event](#)

TBaseVirtualTree.OnDragAllowed Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



TBaseVirtualTree Class |
TBaseVirtualTree.OnDragAllowed
Event |
TBaseVirtualTree.OnDragOver
Event

TBaseVirtualTree.OnDragDrop Event

TBaseVirtualTree Class

Drag'n drop support event.

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.

```
procedure TMainForm.VTDragDrop(Sender: TBaseVirtual
  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 is recommend by Microsoft to order availa
// the first best format which we can accept is
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 TBaseV
          AttachMode := amInsertBefore
        else
          AttachMode := amNowhere;
        end;
        // in the case the drop target does an opti
        // to indicate this also to the drag source
        Sender.ProcessDrop(DataObject, Sender.DropT
        Break;
      end;
    end
  else
    begin
      // VCL drag'n drop, Effects contains by default
      // as usual the application has to find out wha
      Beep;
    end;
  end;
end;

```

Class

TBaseVirtualTree Class

Links

TBaseVirtualTree Class

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnDragDrop](#)
[Event](#) |
[TBaseVirtualTree.OnEditCancelled](#)
[Event](#)

TBaseVirtualTree.OnDragOver Event

[TBaseVirtualTree Class](#) | [See Also](#)

Drag'n drop support event.

Pascal

```
property OnDragOver: TVTDragOverEvent;
```

Description

Triggered when Sender is the potential target of a drag'n drop operation. You can use this event to allow or deny a drop operation by setting Allowed to True or False, respectively. For conditions of OLE or VCL drag source see [OnDragDrop](#).

See Also

[OnDragDrop](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnDragOver](#)
[Event](#) |
[TBaseVirtualTree.OnEdited](#) Event

TBaseVirtualTree.OnEditCancelled Event

[TBaseVirtualTree Class](#) | [See Also](#)

Editing support event.

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnEditCancelled](#)
[Event](#) |
[TBaseVirtualTree.OnEditing Event](#)

TBaseVirtualTree.OnEdited Event

[TBaseVirtualTree Class](#) | [See Also](#)

Editing support event.

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 Class](#) |
[TBaseVirtualTree.OnEdited Event](#)
| [TBaseVirtualTree.OnExpanded Event](#)

TBaseVirtualTree.OnEditing Event

[TBaseVirtualTree Class](#) | [See Also](#)

Editing support event.

Pascal

```
property OnEditing: TVTEditChangingEvent;
```

Description

Triggered when a node is about to be edited. Use **Allowed** to allow or deny this action.

See Also

[Editors and editing](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnEditing Event](#)
| [TBaseVirtualTree.OnExpanding
Event](#)

TBaseVirtualTree.OnExpanded Event

[TBaseVirtualTree Class](#)

Misscellaneous 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 Class](#) |
[TBaseVirtualTree.OnExpanded](#)
[Event](#) |
[TBaseVirtualTree.OnFocusChanged](#)
[Event](#)

TBaseVirtualTree.OnExpanding Event

[TBaseVirtualTree Class](#)

Miscellaneous event.

Pascal

```
property OnExpanding: TVTChangingEvent;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnExpanding](#)
[Event](#) |
[TBaseVirtualTree.OnFocusChanging](#)
[Event](#)

TBaseVirtualTree.OnFocusChanged Event

[TBaseVirtualTree Class](#)

Navigation support event.

Pascal

```
property OnFocusChanged: TVTFocusChangeEvent;
```

Description

Triggered after the focused node changed. When examining **Node** keep in mind that it can be nil, meaning there is no focused node.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnFocusChanged](#)
[Event](#) |
[TBaseVirtualTree.OnFreeNode](#)
[Event](#)

TBaseVirtualTree.OnFocusChanging Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnFocusChanging](#)
[Event](#) |
[TBaseVirtualTree.OnGetCellsEmpty](#)
[Event](#)

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 Class](#) |
[TBaseVirtualTree.OnFreeNode](#)
[Event](#) |
[TBaseVirtualTree.OnGetCursor](#)
[Event](#)

TBaseVirtualTree.OnGetCellsEmpty 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 `OnGetCellsEmpty` to let the application decide.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnGetCellsEmpty](#)
[Event](#) |
[TBaseVirtualTree.OnGetHeaderCursor](#)
[Event](#)

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 Class](#) |
[TBaseVirtualTree.OnGetCursor](#)
[Event](#) |
[TBaseVirtualTree.OnGetHelpContext](#)
[Event](#)

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 Class](#) |
[TBaseVirtualTree.OnGetHeaderCursor](#)
[Event](#) |
[TBaseVirtualTree.OnGetImageIndex](#)
[Event](#)

TBaseVirtualTree.OnGetHelpContext Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnGetHelpContext](#)
[Event](#) |
[TBaseVirtualTree.OnGetImageIndexEx](#)
[Event](#)

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 Class](#) |
[TBaseVirtualTree.OnGetImageIndex](#)
[Event](#) |
[TBaseVirtualTree.OnGetLineStyle](#)
[Event](#)

TBaseVirtualTree.OnGetImageIndexEx Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnGetImageIndexEx](#)
[Event](#) |
[TBaseVirtualTree.OnGetNodeDataSize](#)
[Event](#)

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 `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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnGetLineStyle](#)
[Event](#) |
[TBaseVirtualTree.OnGetPopupMenu](#)
[Event](#)

TBaseVirtualTree.OnGetNodeDataSize Event

[TBaseVirtualTree Class](#) | [See Also](#)

Data management event.

Pascal

```
property OnGetNodeDataSize: TVTGetNodeDataSizeEvent;
```

Description

Triggered when access to a node's data happens the first time but the actual data size is not yet set. Usually you would specify the size of the data you want to have added to each node by [NodeDataSize](#), e.g. `SizeOf(TMyRecord)` is quite usual there (where `TMyRecord` is the structure you want to have stored in the node). Sometimes, however it is not possible to determine the node size in advance, so you can leave [NodeDataSize](#) being `-1` (the default value) and the `OnGetNodeDataSize` event is triggered as soon as the first regular node is created (the hidden root node does not have user data but internal data which is determined by other means).

See Also

[NodeDataSize](#), [Data handling](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class, See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnGetNodeDataSize Event](#) |
[TBaseVirtualTree.OnGetUserClipboardFormats Event](#)

TBaseVirtualTree.OnGetPopupMenu Event

[TBaseVirtualTree Class](#)

Miscellaneous event.

Pascal

```
property OnGetPopupMenu: TVTPopupEvent;
```

Description

This event allows the application to return a popup menu which is specific to a certain node. The tree does an automatic traversal all the way up to the top level node which is the parent of a given node to get a popup menu. If **Menu** is set then the traversal stops. Otherwise it continues until either a menu is set, AskParent is set to False or the top level parent has been reached.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnGetPopupMenu](#)
[Event](#) |
[TBaseVirtualTree.OnHeaderClick](#)
[Event](#)

TBaseVirtualTree.OnGetUserClipboardFormats Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnGetUserClipboardFormats
Event](#) | [TBaseVirtualTree.OnHeaderDbClick
Event](#)

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 Class](#) |
[TBaseVirtualTree.OnHeaderClick](#)
[Event](#) |
[TBaseVirtualTree.OnHeaderDragged](#)
[Event](#)

TBaseVirtualTree.OnHeaderDbClick Event

[TBaseVirtualTree Class](#) | [See Also](#)

Header & column support event.

Pascal

```
property OnHeaderDbClick: 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 Class](#) |
[TBaseVirtualTree.OnHeaderDbClick](#)
[Event](#) |
[TBaseVirtualTree.OnHeaderDraggedOut](#)
[Event](#)

TBaseVirtualTree.OnHeaderDragged Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnHeaderDragged](#)
[Event](#) |
[TBaseVirtualTree.OnHeaderDragging](#)
[Event](#)

TBaseVirtualTree.OnHeaderDraggedOut Event

[TBaseVirtualTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderDraggedOut: TVTHeaderDraggedOutEvent
```

Description

When during a header drag operation the mouse moves out of the header rectangle and the mouse button is released then an OnHeaderDraggedOut event will be fired with the target mouse position in screen coordinates.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnHeaderDraggedOut
Event](#) |
[TBaseVirtualTree.OnHeaderDraw Event](#)

TBaseVirtualTree.OnHeaderDragging Event

[TBaseVirtualTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderDragging: TVTHeaderDraggingEvent;
```

Description

Triggered just before **dragging** of a header button starts. Set **Allowed** to False if you want to prevent the drag operation of the given column.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnHeaderDragging Event](#) |
[TBaseVirtualTree.OnHeaderDrawQueryElements](#)
[Event](#)

TBaseVirtualTree.OnHeaderDraw Event

[TBaseVirtualTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderDraw: TVTHeaderPaintEvent;
```

Description

If you set the hoOwnerDraw style in [TVTHeader.Options](#) and a column has been set to vsOwnerDraw (see also [TVirtualTreeColumn.Style](#)) then OnDrawHeader is called whenever a column needs painting.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnHeaderDraw Event](#) |
[TBaseVirtualTree.OnHeaderMouseDown Event](#)

TBaseVirtualTree.OnHeaderDrawQueryElements Event

[TBaseVirtualTree Class](#) | [See Also](#)

Header & column support event.

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 Class](#) |
[TBaseVirtualTree.OnHeaderDrawQueryElements
Event](#) | [TBaseVirtualTree.OnHeaderMouseMove
Event](#)

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 Class](#) |
[TBaseVirtualTree.OnHeaderMouseDown](#)
Event |
[TBaseVirtualTree.OnHeaderMouseUp](#)
Event

TBaseVirtualTree.OnHeaderMouseMove Event

[TBaseVirtualTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderMouseMove: TVTHeaderMouseMoveEvent;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnHeaderMouseMove
Event](#) | [TBaseVirtualTree.OnHotChange
Event](#)

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 Class](#) |
[TBaseVirtualTree.OnHeaderMouseUp](#)
[Event](#) |
[TBaseVirtualTree.OnIncrementalSearch](#)
[Event](#)

TBaseVirtualTree.OnHotChange Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



TBaseVirtualTree Class |
TBaseVirtualTree.OnHotChange
Event |
TBaseVirtualTree.OnInitChildren
Event

TBaseVirtualTree.OnIncrementalSearch Event

TBaseVirtualTree Class

Miscellaneous event.

Pascal

```
property OnIncrementalSearch: TVTIncrementalSearchEv
```

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  
  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, No
end;

// By using StrLIComp we can specify a maximum le
// which match only partially.
Result := StrLIComp(PChar(S), PChar(PropText), Mi
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 Class](#) |
[TBaseVirtualTree.OnIncrementalSearch
Event](#) | [TBaseVirtualTree.OnInitNode
Event](#)

TBaseVirtualTree.OnInitChildren Event

[TBaseVirtualTree Class](#) | [See Also](#)

Node management event.

Pascal

```
property OnInitChildren: TVTInitChildrenEvent;
```

Description

In order to allow the tree only to fill content where needed it is possible to set the `vsHasChildren` style in a node's 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 Class](#) |
[TBaseVirtualTree.OnInitChildren](#)
[Event](#) |
[TBaseVirtualTree.OnKeyAction](#)
[Event](#)

TBaseVirtualTree.OnInitNode Event

[TBaseVirtualTree Class](#) | [See Also](#)

Node management event.

Pascal

```
property OnInitNode: TVTInitNodeEvent;
```

Description

This event is important to connect the tree to your internal data. It is the ideal place to put references or whatever you need into a node's data area. You can set some initial states like selection, expansion state or that a node has child nodes.

See Also

[The virtual paradigm](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnInitNode](#)
[Event](#) |
[TBaseVirtualTree.OnLoadNode](#)
[Event](#)

TBaseVirtualTree.OnKeyAction Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnKeyAction](#)
[Event](#) |
[TBaseVirtualTree.OnMeasureItem](#)
[Event](#)

TBaseVirtualTree.OnLoadNode Event

[TBaseVirtualTree Class](#) | [See Also](#)

Streaming support event.

Pascal

```
property OnLoadNode: TVTSaveNodeEvent;
```

Description

This event is typically triggered when serialized tree data must be restored, e.g. when loading the tree from file or stream or during a clipboard/drag'n drop operation. You should only read in what you wrote out in [OnSaveNode](#). For safety there is a check in the loader code which tries to keep the internal serialization structure intact in case the application does not read correctly.

See Also

[OnSaveNode](#), [LoadFromStream](#), [SaveToStream](#),
[AddFromStream](#), [VTTTreeStreamVersion](#),
[TVTHeader.LoadFromStream](#), [TVTHeader.SaveToStream](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnLoadNode](#)
[Event](#) |
[TBaseVirtualTree.OnNodeCopied](#)
[Event](#)

TBaseVirtualTree.OnMeasureItem Event

[TBaseVirtualTree Class](#) | [See Also](#)

Miscellaneous event.

Pascal

```
property OnMeasureItem: TVTMeasureItemEvent;
```

Description

Virtual Treeview supports individual node heights. However it might sometimes be unpractical to set this height in advance (e.g. during [OnInitNode](#)). Another scenario might be that multi line nodes must size themselves to accommodate 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 Class](#) |
[TBaseVirtualTree.OnMeasureItem](#)
[Event](#) |
[TBaseVirtualTree.OnNodeCopying](#)
[Event](#)

TBaseVirtualTree.OnNodeCopied Event

[TBaseVirtualTree Class](#)

Miscellaneous event.

Pascal

```
property OnNodeCopied: TVTNodeCopiedEvent;
```

Description

This event is triggered during drag'n drop after a node has been copied to a new location. Sender is the target tree where the copy operation took place.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnNodeCopied](#)
[Event](#) |
[TBaseVirtualTree.OnNodeMoved](#)
[Event](#)

TBaseVirtualTree.OnNodeCopying Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnNodeCopying](#)
[Event](#) |
[TBaseVirtualTree.OnNodeMoving](#)
[Event](#)

TBaseVirtualTree.OnNodeMoved Event

[TBaseVirtualTree Class](#)

Miscellaneous event.

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 Class](#) |
[TBaseVirtualTree.OnNodeMoved](#)
[Event](#) |
[TBaseVirtualTree.OnPaintBackground](#)
[Event](#)

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 Class](#) |
[TBaseVirtualTree.OnNodeMoving](#)
[Event](#) |
[TBaseVirtualTree.OnRenderOLEData](#)
[Event](#)

TBaseVirtualTree.OnPaintBackground Event

[TBaseVirtualTree Class](#)

Paint support event.

Pascal

```
property OnPaintBackground: TVTBackgroundPaintEvent;
```

Description

This event is triggered when the tree has finished its painting and there is an area which is not covered by nodes. For nodes there are various events to allow background customizaton. For the free area in the tree window there is this event.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnPaintBackground](#)
[Event](#) |
[TBaseVirtualTree.OnResetNode](#)
[Event](#)

TBaseVirtualTree.OnRenderOLEData Event

[TBaseVirtualTree 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 descendants) have specified their own formats in [OnGetUserClipboardFormats](#).

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnRenderOLEData](#)
[Event](#) |
[TBaseVirtualTree.OnSaveNode](#)
[Event](#)

TBaseVirtualTree.OnResetNode Event

[TBaseVirtualTree Class](#) | [See Also](#)

Node management event.

Pascal

```
property OnResetNode: TVTChangeEvent;
```

Description

For large trees or simply because the content changed it is sometimes necessary to discard a certain node and release all its children. This can be done with [ResetNode](#) which will trigger this event.

See Also

[ResetNode](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnResetNode
Event](#) | [TBaseVirtualTree.OnScroll
Event](#)

TBaseVirtualTree.OnSaveNode Event

[TBaseVirtualTree 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](#), [VTTStreamVersion](#),
[TVTHHeader.LoadFromStream](#), [TVTHHeader.SaveToStream](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnSaveNode](#)
[Event](#) |
[TBaseVirtualTree.OnShowScrollbar](#)
[Event](#)

TBaseVirtualTree.OnScroll Event

[TBaseVirtualTree Class](#) | [See Also](#)

Miscellaneous event.

Pascal

```
property OnScroll: TVTScrollEvent;
```

Description

This event is triggered when the tree is scrolled horizontally or vertically. You can use it to synchronize scrolling of several trees or other controls.

See Also

[OffsetXY](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnScroll Event](#) |
[TBaseVirtualTree.OnStateChange Event](#)

TBaseVirtualTree.OnShowScrollbar Event

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
property OnShowScrollbar: TVTScrollbarShowEvent;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnShowScrollbar](#)
[Event](#) |
[TBaseVirtualTree.OnStructureChange](#)
[Event](#)

TBaseVirtualTree.OnStateChange Event

[TBaseVirtualTree Class](#)

Miscellaneous event.

Pascal

```
property OnStateChange: TVTStateChangeEvent;
```

Description

For special effects or in order to increase performance it is sometimes useful to know when the tree changes one of its internal states like `tsIncrementalSearching` or `tsOLEDDragging`. 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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnStateChange](#)
[Event](#) |
[TBaseVirtualTree.OnUpdating](#)
[Event](#)

TBaseVirtualTree.OnStructureChange Event

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnStructureChange
Event](#) | [TBaseVirtualTree.RootNode
Property](#)

TBaseVirtualTree.OnUpdating Event

[TBaseVirtualTree Class](#)

Miscellaneous event.

Pascal

```
property OnUpdating: TVTUpdatingEvent;
```

Description

This event is triggered when the application or the tree call [BeginUpdate](#) or [EndUpdate](#) and indicate so when a larger update operation takes place. This can for instance be used to show a hour glass wait cursor.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OnUpdating](#)
[Event](#) |
[TBaseVirtualTree.RootNodeCount](#)
[Property](#)

TBaseVirtualTree.RootNode Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property RootNode: PVirtualNode;
```

Description

For anchoring the tree hierarchy an internal tree node is maintained which is mostly just like any other tree node but has sometimes differently handled. The root node is always expanded and initialized. Its parent member points to the treeview to which the node belongs to and its PreviousSibling and NextSibling members point to the root node itself to make it possible to actually recognize this node.

Notes

You should not use the root node to iterate through the tree. It is only publicly accessible because it is the parent of

all top level nodes and can be used to test a node whether it is a top level node or not.

Class

[TBaseVirtualTree Class](#)

Links

TBaseVirtualTree Class

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.RootNode](#)
[Property](#) |
[TBaseVirtualTree.ScrollBarOptions](#)
[Property](#)

TBaseVirtualTree.RootNodeCount Property

[TBaseVirtualTree Class](#)

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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.RootNodeCount](#)
[Property](#) |
[TBaseVirtualTree.SearchBuffer](#)
[Property](#)

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 Class](#) |
[TBaseVirtualTree.ScrollBarOptions](#)
[Property](#) |
[TBaseVirtualTree.Selected](#)
[Property](#)

TBaseVirtualTree.SearchBuffer Property

[TBaseVirtualTree Class](#) | [See Also](#)

Current input string for incremental search.

Pascal

```
property SearchBuffer: WideString;
```

Description

When incremental search is active you can use SearchBuffer to get the input string typed by the user, which created the last match.

See Also

[IncrementalSearch](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.SearchBuffer](#)
[Property](#) |
[TBaseVirtualTree.SelectedCount](#)
[Property](#)

TBaseVirtualTree.Selected Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property Selected [Node: PVirtualNode]: Boolean;
```

Description

This array property is used to test whether a given node is selected or to switch its selection state. Note that the selection state has nothing to do with the **focused state**. 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 Class](#) |
[TBaseVirtualTree.Selected Property](#) |
[TBaseVirtualTree.SelectionBlendFactor Property](#)

TBaseVirtualTree.SelectedCount Property

[TBaseVirtualTree Class](#)

Contains the number of selected nodes.

Pascal

```
property SelectedCount: Integer;
```

Description

If multiselection is enabled ([toMultiSelect](#)) then [SelectedCount](#) will contain the actual number of selected nodes. In order to change the selection state of a node use [Selected](#) or [AddToSelection/RemoveFromSelection](#).

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.SelectedCount](#)
[Property](#) |
[TBaseVirtualTree.SelectionCurveRadius](#)
[Property](#)

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

```
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 Class](#) |
[TBaseVirtualTree.SelectionBlendFactor](#)
[Property](#) |
[TBaseVirtualTree.StateImages](#)
[Property](#)

TBaseVirtualTree.SelectionCurveRadius Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.SelectionCurveRadius
Property](#) | [TBaseVirtualTree.TextMargin
Property](#)

TBaseVirtualTree.StateImages Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.StateImages](#)
[Property](#) |
[TBaseVirtualTree.TopNode](#)
[Property](#)

TBaseVirtualTree.TextMargin Property

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.TextMargin](#)
Property |
[TBaseVirtualTree.TotalCount](#)
Property

TBaseVirtualTree.TopNode Property

[TBaseVirtualTree Class](#)

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

Pascal

```
property TopNode: PVirtualNode;
```

Description

This property is a reference to the node which is the first node which is at least partially visible in the client area.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



TBaseVirtualTree Class |
TBaseVirtualTree.TopNode Property |
TBaseVirtualTree.TotalInternalDataSize
Property

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 Class](#) |
[TBaseVirtualTree.TotalCount](#)
[Property](#) |
[TBaseVirtualTree.TreeOptions](#)
[Property](#)

TBaseVirtualTree.TotalInternalDataSize Property

[TBaseVirtualTree Class](#) | [See Also](#)

Keeps the currently accumulated data size for one node.

Pascal

```
property TotalInternalDataSize: Cardinal;
```

Description

Each node in the tree not only supports user data but also an 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 Class](#) |
[TBaseVirtualTree.TotalInternalDataSize](#)
[Property](#) | [TBaseVirtualTree.TreeStates](#)
[Property](#)

TBaseVirtualTree.TreeOptions Property

[TBaseVirtualTree Class](#)

Reference to the tree's options.

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 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.TreeOptions](#)
[Property](#) |
[TBaseVirtualTree.UpdateCount](#)
[Property](#)

TBaseVirtualTree.TreeStates Property

[TBaseVirtualTree Class](#) | [See Also](#)

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

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 Class](#) |
[TBaseVirtualTree.TreeStates](#)
[Property](#) |
[TBaseVirtualTree.VerticalAlignment](#)
[Property](#)

TBaseVirtualTree.UpdateCount Property

[TBaseVirtualTree Class](#)

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 Class](#) |
[TBaseVirtualTree.UpdateCount](#)
[Property](#) |
[TBaseVirtualTree.VisibleCount](#)
[Property](#)

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 Class](#) |
[TBaseVirtualTree.VerticalAlignment](#)
[Property](#) |
[TBaseVirtualTree.VisiblePath](#)
[Property](#)

TBaseVirtualTree.VisibleCount Property

[TBaseVirtualTree Class](#)

Number of currently visible nodes.

Pascal

```
property VisibleCount: Cardinal;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.VisibleCount](#)
[Property](#) |
[TBaseVirtualTree.WantTabs](#)
[Property](#)

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 Class](#) |
[TBaseVirtualTree.VisiblePath](#)
[Property](#) |
[TBaseVirtualTree.AbsoluteIndex](#)
[Method](#)

TBaseVirtualTree.WantTabs Property

[TBaseVirtualTree Class](#)

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

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.WantTabs](#)
Property |
[TBaseVirtualTree.AddChild](#)
Method

TBaseVirtualTree.AbsoluteIndex Method

[TBaseVirtualTree Class](#)

Reads the overall index of a node.

Pascal

```
function AbsoluteIndex(Node: PVirtualNode): Cardinal
```

Description

Indicates the index of the tree node relative to the first tree node in a tree.

Notes

Similar to [TotalCount](#) 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 Class](#) |
[TBaseVirtualTree.AbsoluteIndex](#)
[Method](#) |
[TBaseVirtualTree.AddFromStream](#)
[Method](#)

TBaseVirtualTree.AddChild Method

[TBaseVirtualTree Class](#) | [See Also](#)

Creates and adds a new child node to given node.

Pascal

```
function AddChild(Parent: PVirtualNode; UserData: Po
```

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 Class](#) |
[TBaseVirtualTree.AddChild Method](#) |
[TBaseVirtualTree.AdjustPaintCellRect Method](#)

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](#)

What do you think about this topic? [Send feedback!](#)



TBaseVirtualTree Class |
TBaseVirtualTree.AddToSelection
Method (TNodeArray, Integer,
Boolean)

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 Class](#) |
[TBaseVirtualTree.AddToSelection](#)
Method (PVirtualNode)

TBaseVirtualTree.AddToSelection Method (TNodeArray, Integer, Boolean)

[TBaseVirtualTree Class](#)

```
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 Class](#) |
[TBaseVirtualTree.AddFromStream](#)
Method |
[TBaseVirtualTree.AdjustPanningCursor](#)
Method

TBaseVirtualTree.AdjustPaintCellRect Method

[TBaseVirtualTree Class](#)

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

Pascal

```
procedure AdjustPaintCellRect(var PaintInfo: TVTPain
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



TBaseVirtualTree Class |
TBaseVirtualTree.AdjustPaintCellRect
Method |
TBaseVirtualTree.AdviseChangeEvent
Method

TBaseVirtualTree.AdjustPanningCursor Method

TBaseVirtualTree Class

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

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 Class](#) |
[TBaseVirtualTree.AdjustPanningCursor](#)
Method |
[TBaseVirtualTree.AllocateInternalDataArea](#)
Method

TBaseVirtualTree.AdviseChangeEvent Method

[TBaseVirtualTree Class](#)

Used to register a delayed change event.

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 Class](#) |
[TBaseVirtualTree.AdviseChangeEvent
Method](#) | [TBaseVirtualTree.Animate
Method](#)

TBaseVirtualTree.AllocateInternalDataArea Method

[TBaseVirtualTree Class](#) | [See Also](#)

Registration method to allocate tree internal data per node.

Pascal

```
function AllocateInternalDataArea(Size: Cardinal): C
```

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 Class](#) |
[TBaseVirtualTree.AllocateInternalDataArea
Method](#) | [TBaseVirtualTree.Assign Method](#)

TBaseVirtualTree.Animate Method

[TBaseVirtualTree Class](#)

Support method for animated actions in the tree view.

Pascal

```
procedure Animate(Steps: Cardinal; Duration: Cardina
```

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 Class](#) |
[TBaseVirtualTree.Animate Method](#)
| [TBaseVirtualTree.BeginDrag Method](#)

TBaseVirtualTree.Assign Method

[TBaseVirtualTree Class](#)

Used to copy properties from another Virtual Treeview.

Pascal

```
procedure Assign(Source: TPersistent); override;
```

Description

Although this method assigns 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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.Assign Method](#) |
[TBaseVirtualTree.BeginSynch
Method](#)

TBaseVirtualTree.BeginDrag Method

[TBaseVirtualTree Class](#)

Starts an OLE drag'n drop operation.

Pascal

```
procedure BeginDrag(Immediate: Boolean; Threshold: I
```

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 Class](#) |
[TBaseVirtualTree.BeginDrag](#)
Method |
[TBaseVirtualTree.BeginUpdate](#)
Method

TBaseVirtualTree.BeginSynch Method

[TBaseVirtualTree Class](#)

Enters the tree into a special synchronized mode.

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 Class](#) |
[TBaseVirtualTree.BeginSynch Method](#) |
[TBaseVirtualTree.CalculateSelectionRect Method](#)

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 Class](#) |
[TBaseVirtualTree.BeginUpdate](#)
Method |
[TBaseVirtualTree.CanAutoScroll](#)
Method

TBaseVirtualTree.CalculateSelectionRect Method

[TBaseVirtualTree Class](#)

Support method for draw selection.

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 Class](#) |
[TBaseVirtualTree.CalculateSelectionRect](#)
[Method](#) |
[TBaseVirtualTree.CancelCutOrCopy](#)
[Method](#)

TBaseVirtualTree.CanAutoScroll Method

[TBaseVirtualTree Class](#)

Determines whether the tree can currently auto scroll its window.

Pascal

```
function CanAutoScroll: Boolean; virtual;
```

Description

This method was created because the conditions when the tree may automatically scroll its content are quite complex. Additionally, tree descendants might want to add further limitations. Thus the determination has been put into an own method which returns true if the tree is allowed to scroll, otherwise False.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.CanAutoScroll](#)
[Method](#) |
[TBaseVirtualTree.CancelEditNode](#)
[Method](#)

TBaseVirtualTree.CancelCutOrCopy Method

[TBaseVirtualTree Class](#)

Canceles any pending cut or copy clipboard operation.

Pascal

```
procedure CancelCutOrCopy;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.CancelCutOrCopy
Method](#) | [TBaseVirtualTree.CanEdit
Method](#)

TBaseVirtualTree.CancelEditNode Method

[TBaseVirtualTree Class](#)

Cancel the current edit operation, if there is any.

Pascal

```
function CancelEditNode: Boolean;
```

Description

Used to stop the current edit operation. The node editor will get a CancelEdit call so that the node is not changed.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.CancelEditNode](#)
Method |
[TBaseVirtualTree.CanFocus](#)
Method

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 Class](#) |
[TBaseVirtualTree.CanEdit Method](#) |
[TBaseVirtualTree.CanShowDragImage Method](#)

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 Class](#) |
[TBaseVirtualTree.CanFocus](#)
[Method](#) |
[TBaseVirtualTree.Change Method](#)

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 Class](#) |
[TBaseVirtualTree.CanShowDragImage](#)
Method |
[TBaseVirtualTree.ChangeScale](#)
Method

TBaseVirtualTree.Change Method

[TBaseVirtualTree Class](#) | [See Also](#)

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

Pascal

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

Description

The Change method is called to trigger the change 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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.Change Method](#) |
[TBaseVirtualTree.CheckParentCheckState Method](#)

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); overr
```

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 Class](#) |
[TBaseVirtualTree.ChangeScale
Method](#) | [TBaseVirtualTree.Clear
Method](#)

TBaseVirtualTree.CheckParentCheckState Method

[TBaseVirtualTree Class](#)

Helper method for recursive check state changes.

Pascal

```
function CheckParentCheckState(Node: PVirtualNode; N
```

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 Class](#) |
[TBaseVirtualTree.CheckParentCheckState
Method](#) | [TBaseVirtualTree.ClearChecked
Method](#)

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 Class](#) |
[TBaseVirtualTree.Clear Method](#) |
[TBaseVirtualTree.ClearSelection Method](#)

TBaseVirtualTree.ClearChecked Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure ClearChecked;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ClearChecked](#)
Method |
[TBaseVirtualTree.ClearTempCache](#)
Method

TBaseVirtualTree.ClearSelection Method

[TBaseVirtualTree Class](#)

Removes all nodes from the current selection.

Pascal

```
procedure ClearSelection;
```

Description

ClearSelection empties the internal selection cache and resets the vsSelected state from all nodes, which were in this array.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ClearSelection](#)
Method |
[TBaseVirtualTree.ColumnsIsEmpty](#)
Method

TBaseVirtualTree.ClearTempCache Method

[TBaseVirtualTree Class](#)

Helper method to **clear** the internal temporary node cache.

Pascal

```
procedure ClearTempCache; virtual;
```

Description

The internal node cache is used when more than one node is involved in certain operations (e.g. including a range of nodes into the current selection).

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ClearTempCache](#)
Method |
[TBaseVirtualTree.CopyToClipboard](#)
Method

TBaseVirtualTree.ColumnIsEmpty Method

[TBaseVirtualTree Class](#) | [See Also](#)

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

Pascal

```
function ColumnIsEmpty(Node: PVirtualNode; Column: T
```

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](#)

What do you think about this topic? [Send feedback!](#)



TBaseVirtualTree.CopyTo Method (PVirtualNode, PVirtualNode, TVTNodeAttachMode, Boolean)

TBaseVirtualTree Class

Copies **Source** and all its child nodes to **Target**.

Pascal

```
function CopyTo(Source: PVirtualNode; Tree: TBaseVir  
function CopyTo(Source: PVirtualNode; Target: PVirtu
```

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

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ColumnsIsEmpty](#)
Method |
[TBaseVirtualTree.CountLevelDifference](#)
Method

TBaseVirtualTree.CopyToClipboard Method

[TBaseVirtualTree Class](#)

Copies all currently selected nodes to the clipboard.

Pascal

```
procedure CopyToClipboard; virtual;
```

Description

CopyToClipboard causes the tree to copy the currently selected nodes to the clipboard. Actually, Virtual Treeview maintains 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](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.CopyToClipboard](#)
Method |
[TBaseVirtualTree.CountVisibleChildren](#)
Method

TBaseVirtualTree.CountLevelDifference Method

[TBaseVirtualTree Class](#)

Determines the level difference of two nodes.

Pascal

```
function CountLevelDifference(Node1: PVirtualNode; N
```

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 Class](#) |
[TBaseVirtualTree.CountLevelDifference](#)
[Method](#) | [TBaseVirtualTree.Create](#)
[Constructor](#)

TBaseVirtualTree.CountVisibleChildren Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function CountVisibleChildren(Node: PVirtualNode): C
```

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.CountVisibleChildren](#)
Method |
[TBaseVirtualTree.CreateParams](#)
Method

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 Class](#) |
[TBaseVirtualTree.Create](#)
[Constructor](#) |
[TBaseVirtualTree.CreateWnd](#)
[Method](#)

TBaseVirtualTree.CreateParams Method

[TBaseVirtualTree Class](#)

Prepares the creation of the controls window handle.

Pascal

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

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 Class](#) |
[TBaseVirtualTree.CreateParams](#)
[Method](#) |
[TBaseVirtualTree.CutToClipboard](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.CreateWnd](#)
Method |
[TBaseVirtualTree.DefineProperties](#)
Method

TBaseVirtualTree.CutToClipboard Method

[TBaseVirtualTree Class](#)

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

Pascal

```
procedure CutToClipboard; virtual;
```

Description

Similar to [CopyToClipboard](#) 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 Class](#) |
[TBaseVirtualTree.CutToClipboard](#)
Method |
[TBaseVirtualTree.DeleteChildren](#)
Method

TBaseVirtualTree.DefineProperties Method

[TBaseVirtualTree Class](#)

Helper method to customize loading and saving persistent tree data.

Pascal

```
procedure DefineProperties(Filer: TFile); 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 Class](#) |
[TBaseVirtualTree.DefineProperties](#)
[Method](#) |
[TBaseVirtualTree.DeleteNode](#)
[Method](#)

TBaseVirtualTree.DeleteChildren Method

[TBaseVirtualTree Class](#)

Removes all child nodes from the given node.

Pascal

```
procedure DeleteChildren(Node: PVirtualNode; ResetHa
```

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 Class](#) |
[TBaseVirtualTree.DeleteChildren](#)
Method |
[TBaseVirtualTree.DeleteSelectedNodes](#)
Method

TBaseVirtualTree.DeleteNode Method

[TBaseVirtualTree Class](#)

Removes the given node from the tree.

Pascal

```
procedure DeleteNode(Node: PVirtualNode; Reindex: Bo
```

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DeleteNode](#)
[Method](#) |
[TBaseVirtualTree.Destroy](#)
[Destructor](#)

TBaseVirtualTree.DeleteSelectedNodes Method

[TBaseVirtualTree Class](#)

Removes all currently selected nodes form 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 Class](#) |
[TBaseVirtualTree.DeleteSelectedNodes Method](#) |
[TBaseVirtualTree.DetermineHiddenChildrenFlag Method](#)

TBaseVirtualTree.Destroy Destructor

[TBaseVirtualTree Class](#)

Destructor of the control.

Pascal

```
destructor Destroy; override;
```

Description

Frees any allocated data in the tree. All pending operations will be stopped and any remaining node is freed.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) | [TBaseVirtualTree.Destroy Destructor](#) | [TBaseVirtualTree.DetermineHiddenChildrenFlagAllNodes Method](#)

TBaseVirtualTree.DetermineHiddenChildrenFlagAllNodes Method

[TBaseVirtualTree Class](#) | [See Also](#)

Determines whether all children of a given node are hidden.

Pascal

```
procedure DetermineHiddenChildrenFlag(Node: PVirtualTree);
```

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.

DetermineHiddenChildren 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 Class](#) |
[TBaseVirtualTree.DetermineHiddenChildrenFlag](#)
Method |
[TBaseVirtualTree.DetermineHitPositionLTR](#)
Method

TBaseVirtualTree.DetermineHiddenChildrenFlag Method

[TBaseVirtualTree Class](#)

Determines whether all children of all nodes are hidden.

Pascal

```
procedure DetermineHiddenChildrenFlagAllNodes; virtual;
```

Description

As extension to DeterminHiddenChildren this method iteratively determines the hidden children flag for all existing nodes in the tree. This is only used for large updates. No node will be initialized in this process.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DetermineHiddenChildrenFlagAllNodes
Method](#) | [TBaseVirtualTree.DetermineNextCheckState
Method](#)

TBaseVirtualTree.DetermineHitPositionLT Method

[TBaseVirtualTree Class](#)

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

Pascal

```
procedure DetermineHitPositionLTR(var HitInfo: THitI  
procedure DetermineHitPositionRTL(var HitInfo: THitI
```

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 Class](#) |
[TBaseVirtualTree.DetermineHitPositionLTR](#)
[Method](#) |
[TBaseVirtualTree.DetermineScrollDirections](#)
[Method](#)

TBaseVirtualTree.DetermineNextCheckState Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DetermineNextCheckState(CheckType: TCheckTy
```

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 Class](#) |
[TBaseVirtualTree.DetermineNextCheckState](#)
Method |
[TBaseVirtualTree.DoAdvancedHeaderDraw](#)
Method

TBaseVirtualTree.DetermineScrollDirection Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DetermineScrollDirections(X: Integer; Y: In
```

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 Class](#) |
[TBaseVirtualTree.DetermineScrollDirections](#)
[Method](#) | [TBaseVirtualTree.DoAfterCellPaint](#)
[Method](#)

TBaseVirtualTree.DoAdvancedHeaderDraw Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoAdvancedHeaderDraw(var PaintInfo: THeade
```

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 Class](#) |
[TBaseVirtualTree.DoAdvancedHeaderDraw](#)
[Method](#) |
[TBaseVirtualTree.DoAfterItemErase](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.DoAfterCellPaint](#)
Method |
[TBaseVirtualTree.DoAfterItemPaint](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoAfterItemErase](#)
Method |
[TBaseVirtualTree.DoAfterPaint](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoAfterItemPaint](#)
Method |
[TBaseVirtualTree.DoAutoScroll](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoAfterPaint](#)
Method |
[TBaseVirtualTree.DoBeforeCellPaint](#)
Method

TBaseVirtualTree.DoAutoScroll Method

[TBaseVirtualTree Class](#)

Enables or disables the auto scroll timer.

Pascal

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

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 Class](#) |
[TBaseVirtualTree.DoAutoScroll](#)
Method |
[TBaseVirtualTree.DoBeforeDrag](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoBeforeCellPaint](#)
Method |
[TBaseVirtualTree.DoBeforeItemErase](#)
Method

TBaseVirtualTree.DoBeforeDrag Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoBeforeDrag(Node: PVirtualNode; Column: TC
```

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 Class](#) |
[TBaseVirtualTree.DoBeforeDrag](#)
Method |
[TBaseVirtualTree.DoBeforeItemPaint](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoBeforeItemErase](#)
Method |
[TBaseVirtualTree.DoBeforePaint](#)
Method

TBaseVirtualTree.DoBeforeItemPaint Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.DoBeforeItemPaint](#)
Method |
[TBaseVirtualTree.DoCancelEdit](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoBeforePaint](#)
Method |
[TBaseVirtualTree.DoCanEdit](#)
Method

TBaseVirtualTree.DoCancelEdit Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function DoCancelEdit: Boolean; virtual;
```

Description

This method calls the edit link's `IEditLink.CancelEdit` method and stops the edit mode if this call returns `True`. If stopping is allowed then the event `OnEditCancelled` 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 Class](#) |
[TBaseVirtualTree.DoCancelEdit](#)
Method |
[TBaseVirtualTree.DoChange](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoCanEdit](#)
Method |
[TBaseVirtualTree.DoCheckClick](#)
Method

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoChange](#)
Method |
[TBaseVirtualTree.DoChecked](#)
Method

TBaseVirtualTree.DoCheckClick Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoCheckClick(Node: PVirtualNode; NewChecksS
```

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 Class](#) |
[TBaseVirtualTree.DoCheckClick](#)
Method |
[TBaseVirtualTree.DoChecking](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoChecked](#)
Method |
[TBaseVirtualTree.DoCollapsed](#)
Method

TBaseVirtualTree.DoChecking Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoChecking(Node: PVirtualNode; var NewCheck
```

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 Class](#) |
[TBaseVirtualTree.DoChecking](#)
Method |
[TBaseVirtualTree.DoCollapsing](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoCollapsed](#)
Method |
[TBaseVirtualTree.DoColumnClick](#)
Method

TBaseVirtualTree.DoCollapsing Method

[TBaseVirtualTree Class](#)

Not documented.

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoCollapsing](#)
[Method](#) |
[TBaseVirtualTree.DoColumnDbClick](#)
[Method](#)

TBaseVirtualTree.DoColumnClick Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoColumnClick(Column: TColumnIndex; Shift:
```

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 Class](#) |
[TBaseVirtualTree.DoColumnClick](#)
Method |
[TBaseVirtualTree.DoColumnResize](#)
Method

TBaseVirtualTree.DoColumnDbClick Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoColumnDbClick(Column: TColumnIndex; Shi
```

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 Class](#) |
[TBaseVirtualTree.DoColumnDbClick](#)
Method |
[TBaseVirtualTree.DoCompare](#)
Method

TBaseVirtualTree.DoColumnResize Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoColumnResize(Column: TColumnIndex); virt
```

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 Class](#) |
[TBaseVirtualTree.DoColumnResize](#)
Method |
[TBaseVirtualTree.DoCreateDataObject](#)
Method

TBaseVirtualTree.DoCompare Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoCompare(Node1: PVirtualNode; Node2: PVirt
```

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 Class](#) |
[TBaseVirtualTree.DoCompare Method](#) |
[TBaseVirtualTree.DoCreateDragManager Method](#)

TBaseVirtualTree.DoCreateDataObject Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoCreateDataObject: IDataObject; virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoCreateDataObject](#)
[Method](#) |
[TBaseVirtualTree.DoCreateEditor](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.DoCreateDragManager
Method](#) | [TBaseVirtualTree.DoDragDrop
Method](#)

TBaseVirtualTree.DoCreateEditor Method

[TBaseVirtualTree 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

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoCreateEditor](#)
Method |
[TBaseVirtualTree.DoDragExpand](#)
Method

TBaseVirtualTree.DoDragDrop Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoDragDrop(Source: TObject; DataObject: ID
```

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 Class](#) |
[TBaseVirtualTree.DoDragDrop](#)
Method |
[TBaseVirtualTree.DoDragging](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoDragExpand](#)
Method |
[TBaseVirtualTree.DoDragOver](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoDragging](#)
[Method](#) | [TBaseVirtualTree.DoEdit](#)
[Method](#)

TBaseVirtualTree.DoDragOver Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoDragOver(Source: TObject; Shift: TShiftSt
```

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 Class](#) |
[TBaseVirtualTree.DoDragOver](#)
Method |
[TBaseVirtualTree.DoEndDrag](#)
Method

TBaseVirtualTree.DoEdit Method

[TBaseVirtualTree Class](#) | [See Also](#)

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

Pascal

```
procedure DoEdit; virtual;
```

Description

This method takes care for editor creation and initialization. You can look for `tsEditing` in [TreeStates](#) 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 Class](#) |
[TBaseVirtualTree.DoEdit Method](#) |
[TBaseVirtualTree.DoEndEdit Method](#)

TBaseVirtualTree.DoEndDrag Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

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

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 Class](#) |
[TBaseVirtualTree.DoEndDrag](#)
Method |
[TBaseVirtualTree.DoExpanded](#)
Method

TBaseVirtualTree.DoEndEdit Method

[TBaseVirtualTree Class](#) | [See Also](#)

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

Pascal

```
function DoEndEdit: Boolean; virtual;
```

Description

The method also sends a message to the tree window to asynchronously release the edit link which communicates to the actual editor. The edit link is responsible to propagate any changes made in its node editor to the tree.

Notes

[TVirtualStringTree](#) 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 Class](#) |
[TBaseVirtualTree.DoEndEdit](#)
Method |
[TBaseVirtualTree.DoExpanding](#)
Method

TBaseVirtualTree.DoExpanded Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoExpanded(Node: PVirtualNode); virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoExpanded](#)
Method |
[TBaseVirtualTree.DoFocusChange](#)
Method

TBaseVirtualTree.DoExpanding Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoExpanding(Node: PVirtualNode): Boolean; v
```

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 Class](#) |
[TBaseVirtualTree.DoExpanding](#)
Method |
[TBaseVirtualTree.DoFocusChanging](#)
Method

TBaseVirtualTree.DoFocusChange Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.DoFocusChange](#)
Method |
[TBaseVirtualTree.DoFocusNode](#)
Method

TBaseVirtualTree.DoFocusChanging Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoFocusChanging(OldNode: PVirtualNode; NewN
```

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 Class](#) |
[TBaseVirtualTree.DoFocusChanging](#)
Method |
[TBaseVirtualTree.DoFreeNode](#)
Method

TBaseVirtualTree.DoFocusNode Method

[TBaseVirtualTree Class](#)

Internal method to set the focused node.

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 Class](#) |
[TBaseVirtualTree.DoFocusNode](#)
Method |
[TBaseVirtualTree.DoGetAnimationType](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoFreeNode](#)
[Method](#) |
[TBaseVirtualTree.DoGetCursor](#)
[Method](#)

TBaseVirtualTree.DoGetAnimationType Method

[TBaseVirtualTree Class](#)

Determines the type of animation to be used.

Pascal

```
function DoGetAnimationType: THintAnimationType; var
```

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. Additionally, 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 Class](#) |
[TBaseVirtualTree.DoGetAnimationType](#)
Method |
[TBaseVirtualTree.DoGetHeaderCursor](#)
Method

TBaseVirtualTree.DoGetCursor Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoGetCursor(var Cursor: TCursor); virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoGetCursor](#)
Method |
[TBaseVirtualTree.DoGetImageIndex](#)
Method

TBaseVirtualTree.DoGetHeaderCursor Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoGetHeaderCursor(var Cursor: HCURSOR); vi
```

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 Class](#) |
[TBaseVirtualTree.DoGetHeaderCursor](#)
Method |
[TBaseVirtualTree.DoGetLineStyle](#)
Method

TBaseVirtualTree.DoGetImageIndex Method

[TBaseVirtualTree Class](#)

Not documented.

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoGetImageIndex](#)
Method |
[TBaseVirtualTree.DoGetNodeHint](#)
Method

TBaseVirtualTree.DoGetLineStyle Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoGetLineStyle(var Bits: Pointer); virtual
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoGetLineStyle](#)
[Method](#) |
[TBaseVirtualTree.DoGetNodeTooltip](#)
[Method](#)

TBaseVirtualTree.DoGetNodeHint Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoGetNodeHint(Node: PVirtualNode; Column: 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 Class](#) |
[TBaseVirtualTree.DoGetNodeHint](#)
Method |
[TBaseVirtualTree.DoGetNodeWidth](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoGetNodeTooltip](#)
Method |
[TBaseVirtualTree.DoGetPopupMenu](#)
Method

TBaseVirtualTree.DoGetNodeWidth Method

[TBaseVirtualTree Class](#)

Overridable method which always returns 0.

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 Class](#) |
[TBaseVirtualTree.DoGetNodeWidth Method](#) |
[TBaseVirtualTree.DoGetUserClipboardFormats Method](#)

TBaseVirtualTree.DoGetPopupMenu Method

[TBaseVirtualTree Class](#)

Overridable method which triggers the OnGetPopup event.

Pascal

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

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoGetPopupMenu](#)
Method |
[TBaseVirtualTree.DoHeaderClick](#)
Method

TBaseVirtualTree.DoGetUserClipboardFormats Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoGetUserClipboardFormats(var Formats: TFo
```

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 Class](#) |
[TBaseVirtualTree.DoGetUserClipboardFormats
Method](#) | [TBaseVirtualTree.DoHeaderDbClick
Method](#)

TBaseVirtualTree.DoHeaderClick Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoHeaderClick(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 Class](#) |
[TBaseVirtualTree.DoHeaderClick](#)
Method |
[TBaseVirtualTree.DoHeaderDragged](#)
Method

TBaseVirtualTree.DoHeaderDbClick Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoHeaderDbClick(Column: TColumnIndex; But
```

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 Class](#) |
[TBaseVirtualTree.DoHeaderDbClick](#)
Method |
[TBaseVirtualTree.DoHeaderDraggedOut](#)
Method

TBaseVirtualTree.DoHeaderDragged Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoHeaderDragged(Column: TColumnIndex; OldP
```

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 Class](#) |
[TBaseVirtualTree.DoHeaderDragged](#)
Method |
[TBaseVirtualTree.DoHeaderDragging](#)
Method

TBaseVirtualTree.DoHeaderDraggedOut Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoHeaderDraggedOut(Column: TColumnIndex; D
```

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 Class](#) |
[TBaseVirtualTree.DoHeaderDraggedOut](#)
[Method](#) |
[TBaseVirtualTree.DoHeaderDraw](#)
[Method](#)

TBaseVirtualTree.DoHeaderDragging Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.DoHeaderDragging Method](#) |
[TBaseVirtualTree.DoHeaderDrawQueryElements Method](#)

TBaseVirtualTree.DoHeaderDraw Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoHeaderDraw(Canvas: TCanvas; Column: TVir
```

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 Class](#) |
[TBaseVirtualTree.DoHeaderDraw](#)
[Method](#) |
[TBaseVirtualTree.DoHeaderMouseDown](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.DoHeaderDrawQueryElements](#)
[Method](#) |
[TBaseVirtualTree.DoHeaderMouseMove Method](#)

TBaseVirtualTree.DoHeaderMouseDown Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoHeaderMouseDown(Button: TMouseButton; Sh
```

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 Class](#) |
[TBaseVirtualTree.DoHeaderMouseDown](#)
Method |
[TBaseVirtualTree.DoHeaderMouseUp](#)
Method

TBaseVirtualTree.DoHeaderMouseMove Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoHeaderMouseMove(Shift: TShiftState; X: I
```

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 Class](#) |
[TBaseVirtualTree.DoHeaderMouseMove](#)
[Method](#) |
[TBaseVirtualTree.DoHotChange Method](#)

TBaseVirtualTree.DoHeaderMouseUp Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoHeaderMouseUp(Button: TMouseButton; Shif
```

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 Class](#) |
[TBaseVirtualTree.DoHeaderMouseUp](#)
Method |
[TBaseVirtualTree.DoIncrementalSearch](#)
Method

TBaseVirtualTree.DoHotChange Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

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

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 Class](#) |
[TBaseVirtualTree.DoHotChange](#)
Method |
[TBaseVirtualTree.DoInitChildren](#)
Method

TBaseVirtualTree.DoIncrementalSearch Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoIncrementalSearch(Node: PVirtualNode; con
```

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 Class](#) |
[TBaseVirtualTree.DoIncrementalSearch
Method](#) | [TBaseVirtualTree.DoInitNode
Method](#)

TBaseVirtualTree.DoInitChildren Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoInitChildren(Node: PVirtualNode; var Chi
```

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 Class](#) |
[TBaseVirtualTree.DoInitChildren](#)
Method |
[TBaseVirtualTree.DoKeyAction](#)
Method

TBaseVirtualTree.DoInitNode Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

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

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 Class](#) |
[TBaseVirtualTree.DoInitNode](#)
Method |
[TBaseVirtualTree.DoLoadUserData](#)
Method

TBaseVirtualTree.DoKeyAction Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoKeyAction(var CharCode: Word; var Shift:
```

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 Class](#) |
[TBaseVirtualTree.DoKeyAction](#)
Method |
[TBaseVirtualTree.DoMeasureItem](#)
Method

TBaseVirtualTree.DoLoadUserData Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.DoLoadUserData](#)
Method |
[TBaseVirtualTree.DoNodeCopied](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoMeasureItem](#)
Method |
[TBaseVirtualTree.DoNodeCopying](#)
Method

TBaseVirtualTree.DoNodeCopied Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoNodeCopied(Node: PVirtualNode); virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoNodeCopied](#)
Method |
[TBaseVirtualTree.DoNodeMoved](#)
Method

TBaseVirtualTree.DoNodeCopying Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.DoNodeCopying](#)
Method |
[TBaseVirtualTree.DoNodeMoving](#)
Method

TBaseVirtualTree.DoNodeMoved Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoNodeMoved(Node: PVirtualNode); virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoNodeMoved](#)
Method |
[TBaseVirtualTree.DoPaintBackground](#)
Method

TBaseVirtualTree.DoNodeMoving Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoNodeMoving(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 Class](#) |
[TBaseVirtualTree.DoNodeMoving](#)
Method |
[TBaseVirtualTree.DoPaintDropMark](#)
Method

TBaseVirtualTree.DoPaintBackground Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoPaintBackground(Canvas: TCanvas; R: TRect
```

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 Class](#) |
[TBaseVirtualTree.DoPaintBackground](#)
Method |
[TBaseVirtualTree.DoPaintNode](#)
Method

TBaseVirtualTree.DoPaintDropMark Method

[TBaseVirtualTree Class](#)

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

Pascal

```
procedure DoPaintDropMark(Canvas: TCanvas; Node: PVi
```

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 Class](#) |
[TBaseVirtualTree.DoPaintDropMark](#)
[Method](#) |
[TBaseVirtualTree.DoPopupMenu](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.DoPaintNode](#)
Method |
[TBaseVirtualTree.DoRenderOLEData](#)
Method

TBaseVirtualTree.DoPopupMenu Method

[TBaseVirtualTree Class](#)

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

Pascal

```
procedure DoPopupMenu(Node: PVirtualNode; Column: TC
```

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 Class](#) |
[TBaseVirtualTree.DoPopupMenu](#)
[Method](#) |
[TBaseVirtualTree.DoReset Method](#)

TBaseVirtualTree.DoRenderOLEData Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoRenderOLEData(const FormatEtcIn: TFormatE
```

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 Class](#) |
[TBaseVirtualTree.DoRenderOLEData](#)
Method |
[TBaseVirtualTree.DoSaveUserData](#)
Method

TBaseVirtualTree.DoReset Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoReset(Node: PVirtualNode); virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoReset Method](#)
| [TBaseVirtualTree.DoScroll Method](#)

TBaseVirtualTree.DoSaveUserData Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoSaveUserData(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 Class](#) |
[TBaseVirtualTree.DoSaveUserData](#)
Method |
[TBaseVirtualTree.DoSetOffsetXY](#)
Method

TBaseVirtualTree.DoScroll Method

[TBaseVirtualTree Class](#)

Overridable method which triggers the [OnScroll](#) event.

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 Class](#) |
[TBaseVirtualTree.DoScroll Method](#) |
[TBaseVirtualTree.DoShowScrollbar Method](#)

TBaseVirtualTree.DoSetOffsetXY Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function DoSetOffsetXY(Value: TPoint; Options: TScro
```

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 Class](#) |
[TBaseVirtualTree.DoSetOffsetXY](#)
Method |
[TBaseVirtualTree.DoStartDrag](#)
Method

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 Class](#) |
[TBaseVirtualTree.DoShowScrollbar](#)
Method |
[TBaseVirtualTree.DoStateChange](#)
Method

TBaseVirtualTree.DoStartDrag Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.DoStartDrag](#)
Method |
[TBaseVirtualTree.DoStructureChange](#)
Method

TBaseVirtualTree.DoStateChange Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoStateChange(Enter: TVirtualTreeStates; L
```

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 Class](#) |
[TBaseVirtualTree.DoStateChange](#)
Method |
[TBaseVirtualTree.DoTimerScroll](#)
Method

TBaseVirtualTree.DoStructureChange Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoStructureChange(Node: PVirtualNode; Reas
```

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 Class](#) |
[TBaseVirtualTree.DoStructureChange](#)
Method |
[TBaseVirtualTree.DoUpdating](#)
Method

TBaseVirtualTree.DoTimerScroll Method

[TBaseVirtualTree Class](#)

Callback method which is triggered whenever the scroll timer fires.

Pascal

```
procedure DoTimerScroll; virtual;
```

Description

This method is called to do an automatic tree scroll when the user selects nodes with the mouse (multiselection only).

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoTimerScroll](#)
Method |
[TBaseVirtualTree.DoValidateCache](#)
Method

TBaseVirtualTree.DoUpdating Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DoUpdating(State: TVTUpdateState); virtual
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoUpdating](#)
Method |
[TBaseVirtualTree.DragCanceled](#)
Method

TBaseVirtualTree.DoValidateCache Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DoValidateCache: Boolean; virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DoValidateCache](#)
[Method](#) |
[TBaseVirtualTree.DragDrop](#)
[Method](#)

TBaseVirtualTree.DragCanceled Method

[TBaseVirtualTree Class](#)

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

Pascal

```
procedure DragCanceled; override;
```

Description

DragCanceled is used to do some housekeeping in the tree.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DragCanceled](#)
Method |
[TBaseVirtualTree.DragEnter](#)
Method

TBaseVirtualTree.DragDrop Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function DragDrop(const DataObject: IDataObject; Key
```

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 Class](#) |
[TBaseVirtualTree.DragDrop](#)
Method |
[TBaseVirtualTree.DragFinished](#)
Method

TBaseVirtualTree.DragEnter Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DragEnter(KeyState: Integer; Pt: TPoint; va
```

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 Class](#) |
[TBaseVirtualTree.DragEnter](#)
Method |
[TBaseVirtualTree.Dragging](#)
Method

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 Class](#) |
[TBaseVirtualTree.DragFinished](#)
Method |
[TBaseVirtualTree.DragLeave](#)
Method

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 Class](#) |
[TBaseVirtualTree.Dragging](#)
[Method](#) |
[TBaseVirtualTree.DragOver](#)
[Method](#)

TBaseVirtualTree.DragLeave Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DragLeave; virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.DragLeave](#)
Method |
[TBaseVirtualTree.DrawDottedHLine](#)
Method

TBaseVirtualTree.DragOver Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function DragOver(Source: TObject; KeyState: 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 Class](#) |
[TBaseVirtualTree.DragOver](#)
Method |
[TBaseVirtualTree.DrawDottedVLine](#)
Method

TBaseVirtualTree.DrawDottedHLine Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DrawDottedHLine(const PaintInfo: TVTPaintI
```

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 Class](#) |
[TBaseVirtualTree.DrawDottedHLine](#)
[Method](#) |
[TBaseVirtualTree.EditNode](#) Method

TBaseVirtualTree.DrawDottedVLine Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure DrawDottedVLine(const PaintInfo: TVTPaintI
```

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 Class](#) |
[TBaseVirtualTree.DrawDottedVLine](#)
[Method](#) |
[TBaseVirtualTree.EndEditNode](#)
[Method](#)

TBaseVirtualTree.EditNode Method

[TBaseVirtualTree Class](#) | [See Also](#)

Starts editing the given node if allowed to.

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 Class](#) |
[TBaseVirtualTree.EditNode](#)
[Method](#) |
[TBaseVirtualTree.EndSynch](#)
[Method](#)

TBaseVirtualTree.EndEditNode Method

[TBaseVirtualTree Class](#) | [See Also](#)

Stops node editing if it was started before.

Pascal

```
function EndEditNode: Boolean;
```

Description

EndEditNode stops node editing and accepts the result (which must be set by the edit link).

See Also

[Editors and editing](#), [EditNode](#), [DoEdit](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.EndEditNode](#)
Method |
[TBaseVirtualTree.EndUpdate](#)
Method

TBaseVirtualTree.EndSynch Method

[TBaseVirtualTree Class](#) | [See Also](#)

Counterpart to [BeginSynch](#).

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 Class](#) |
[TBaseVirtualTree.EndSynch](#)
[Method](#) |
[TBaseVirtualTree.ExecuteAction](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.EndUpdate Method](#) |
[TBaseVirtualTree.FindNodeInSelection Method](#)

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 Class](#) |
[TBaseVirtualTree.ExecuteAction](#)
Method |
[TBaseVirtualTree.FinishChunkHeader](#)
Method

TBaseVirtualTree.FindNodeInSelection Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function FindNodeInSelection(P: PVirtualNode; var In
```

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 Class](#) |
[TBaseVirtualTree.FindNodeInSelection](#)
Method |
[TBaseVirtualTree.FinishCutOrCopy](#)
Method

TBaseVirtualTree.FinishChunkHeader Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure FinishChunkHeader(Stream: TStream; StartPo
```

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 Class](#) |
[TBaseVirtualTree.FinishChunkHeader](#)
[Method](#) |
[TBaseVirtualTree.FlushClipboard](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.FinishCutOrCopy](#)
[Method](#) |
[TBaseVirtualTree.FontChanged](#)
[Method](#)

TBaseVirtualTree.FlushClipboard Method

[TBaseVirtualTree Class](#)

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 Class](#) |
[TBaseVirtualTree.FlushClipboard](#)
Method |
[TBaseVirtualTree.FullCollapse](#)
Method

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 Class](#) |
[TBaseVirtualTree.FontChanged](#)
Method |
[TBaseVirtualTree.FullExpand](#)
Method

TBaseVirtualTree.FullCollapse Method

[TBaseVirtualTree Class](#) | [See Also](#)

Collapses all nodes in the tree.

Pascal

```
procedure FullCollapse(Node: PVirtualNode = nil); vi
```

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 Class](#) |
[TBaseVirtualTree.FullCollapse Method](#) |
[TBaseVirtualTree.GetBorderDimensions Method](#)

TBaseVirtualTree.FullExpand Method

[TBaseVirtualTree Class](#)

Expands all nodes in the tree.

Pascal

```
procedure FullExpand(Node: PVirtualNode = nil); virt
```

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 Class](#) |
[TBaseVirtualTree.FullExpand](#)
[Method](#) |
[TBaseVirtualTree.GetCheckImage](#)
[Method](#)

TBaseVirtualTree.GetBorderDimensions Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function GetBorderDimensions: TSize; virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetBorderDimensions](#)
Method |
[TBaseVirtualTree.GetCheckImageListFor](#)
Method

TBaseVirtualTree.GetCheckImage Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.GetCheckImage](#)
Method |
[TBaseVirtualTree.GetColumnClass](#)
Method

TBaseVirtualTree.GetCheckImageListFor Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
class function GetCheckImageListFor(Kind: TCheckImag
```

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 Class](#) |
[TBaseVirtualTree.GetCheckImageListFor](#)
Method |
[TBaseVirtualTree.GetControlsAlignment](#)
Method

TBaseVirtualTree.GetColumnClass Method

[TBaseVirtualTree Class](#)

Returns the class to be used to manage columns in the tree.

Pascal

```
function GetColumnClass: TVirtualTreeColumnClass; vi
```

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 descendants can override this method to return own classes.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetColumnClass](#)
Method |
[TBaseVirtualTree.GetDisplayRect](#)
Method

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 Class |
TBaseVirtualTree.GetControlsAlignment
Method | TBaseVirtualTree.GetFirst
Method

TBaseVirtualTree.GetDisplayRect Method

TBaseVirtualTree Class

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

Pascal

```
function GetDisplayRect(Node: PVirtualNode; Column:
```

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

```
function GetFirst: PVirtualNode;  
function GetFirstChild(Node: PVirtualNode): PVirtual  
function GetFirstCutCopy: PVirtualNode;  
function GetFirstInitialized: PVirtualNode;  
function GetFirstNoInit: PVirtualNode;  
function GetFirstSelected: PVirtualNode;  
function GetFirstVisible: PVirtualNode;  
function GetFirstVisibleChild(Node: PVirtualNode): P  
function GetFirstVisibleChildNoInit(Node: PVirtualNo  
function GetFirstVisibleNoInit: PVirtualNode;
```

Description

This group of navigation functions is used to return the first node in the tree or first sub node with various properties.

GetFirst	First node in the tree with initialization.
GetFirstChild	First child node with initialization.
GetFirstCutCopy	First node in cut/copy set (no initialization needed).
GetFirstInitialized	First initialized node in the tree (no initialization needed).
GetFirstNoInit	First node in the tree without

	initialization.
GetFirstVisible	First visible node in the tree with initialization.
GetFirstVisibleChild	First visible child of a node with initialization.
GetFirstVisibleChildNoInit	First visible child of a node without initialization.
GetFirstVisibleNoInit	First visible node in the tree without initialization.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetFirst Method](#) |
[TBaseVirtualTree.GetHeaderClass Method](#)

TBaseVirtualTree.GetFirstChecked Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function GetFirstChecked(State: TCheckState): PVirtu
```

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 Class](#) |
[TBaseVirtualTree.GetFirstChecked](#)
Method |
[TBaseVirtualTree.GetHintWindowClass](#)
Method

TBaseVirtualTree.GetHeaderClass Method

[TBaseVirtualTree Class](#)

Returns the header class to be used by the tree.

Pascal

```
function GetHeaderClass: TVTHeaderClass; virtual;
```

Description

As with several other classes in Virtual Treeview (e.g. drag manager, options etc.) also a customized header class is supported, which allows applications or descendant classes to implement their very own header class with special behavior. This is a further element to make Virtual Treeview as flexible as possible.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetHeaderClass](#)
Method |
[TBaseVirtualTree.GetHitTestInfoAt](#)
Method

TBaseVirtualTree.GetHintWindowClass Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function GetHintWindowClass: THintWindowClass; virtu
```

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 Class](#) |
[TBaseVirtualTree.GetHintWindowClass](#)
[Method](#) |
[TBaseVirtualTree.GetImageIndex](#)
[Method](#)

TBaseVirtualTree.GetHitTestInfoAt Method

[TBaseVirtualTree Class](#)

Returns information about the node at the given position.

Pascal

```
procedure GetHitTestInfoAt(X: Integer; Y: Integer; R
```

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 Class](#) |
[TBaseVirtualTree.GetHitTestInfoAt](#)
[Method](#) |
[TBaseVirtualTree.GetLast Method](#)

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

```
function GetLast(Node: PVirtualNode = nil): PVirtualNode;
function GetLastInitialized(Node: PVirtualNode = nil): PVirtualNode;
function GetLastNoInit(Node: PVirtualNode = nil): PVirtualNode;
function GetLastChild(Node: PVirtualNode): PVirtualNode;
function GetLastChildNoInit(Node: PVirtualNode): PVirtualNode;
function GetLastVisible(Node: PVirtualNode = nil): PVirtualNode;
function GetLastVisibleChild(Node: PVirtualNode): PVirtualNode;
function GetLastVisibleChildNoInit(Node: PVirtualNode): PVirtualNode;
function GetLastVisibleNoInit(Node: PVirtualNode = nil): PVirtualNode;
```

Description

This group of navigation functions is used to return the last node in the tree or last sub node with various properties.

GetLast	Last node in the tree with initialization.
GetLastChild	Last child node with initialization.
GetLastChildNoInit	Last child node without initialization.
GetLastInitialized	Last initialized node in the tree (no initialization needed).
GetLastNoInit	Last node in the tree without initialization.
	Last visible node in the tree with

GetLastVisible	initialization.
GetLastVisibleChild	Last visible child of a node with initialization.
GetLastVisibleChildNoInit	Last visible child of a node without initialization.
GetLastVisibleNoInit	Last visible node in the tree without initialization.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetLast Method](#) |
[TBaseVirtualTree.GetMaxRightExtend Method](#)

TBaseVirtualTree.GetMaxColumnWidth Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function GetMaxColumnWidth(Column: TColumnIndex): In
```

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 Class](#) |
[TBaseVirtualTree.GetMaxColumnWidth](#)
[Method](#) |
[TBaseVirtualTree.GetNativeClipboardFormats](#)
[Method](#)

TBaseVirtualTree.GetMaxRightExtend Method

[TBaseVirtualTree Class](#)

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

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetMaxRightExtend
Method](#) | [TBaseVirtualTree.GetNext
Method](#)

TBaseVirtualTree.GetNativeClipboardForm Method

[TBaseVirtualTree Class](#)

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

Pascal

```
procedure GetNativeClipboardFormats(var Formats: TFo
```

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.

GetNext	Next node in the tree with initialization.
GetNextCutCopy	Next node in the cut/copy set (no initialization needed).
GetNextInitialized	Next initialized node in the tree (no initialization needed).
GetNextNoInit	Next node in the tree without initialization.

GetNextSelected	Next selected node (no initialization needed).
GetNextSibling	Next sibling node with initialization.
GetNextVisible	Next visible node in the tree with initialization.
GetNextVisibleNoInit	Next visible node in the tree without initialization.
GetNextVisibleSibling	Next visible sibling node with initialization.
GetNextVisibleSiblingNoInit	Next visible sibling node without initialization.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetNext Method](#)
| [TBaseVirtualTree.GetNodeData Method](#)

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 Class |
TBaseVirtualTree.GetNodeAt
Method (Integer, Integer, Boolean,
Integer)

TBaseVirtualTree.GetNodeAt Method (Integer, Integer)

TBaseVirtualTree Class

Not documented.

Pascal

```
function GetNodeAt(X: Integer; Y: Integer): 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 Class](#) |
[TBaseVirtualTree.GetNodeAt
Method \(Integer, Integer\)](#)

TBaseVirtualTree.GetNodeAt Method (Integer, Integer, Boolean, Integer)

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function GetNodeAt(X: Integer; Y: Integer; Relative:
```

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 Class](#) |
[TBaseVirtualTree.GetNextChecked](#)
Method |
[TBaseVirtualTree.GetNodeLevel](#)
Method

TBaseVirtualTree.GetNodeData Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function GetNodeData(Node: PVirtualNode): Pointer;
```

Description

GetNodeData returns the address of the user data area for **Node**. It is strongly recommended to use this method instead directly accessing @Node.Data. Some trees require internal data for their own use which is also stored after Node.Data and the actual user data (application data) follows then this internal data. GetNodeData takes care of this situation.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetNodeData](#)
Method |
[TBaseVirtualTree.GetOptionsClass](#)
Method

TBaseVirtualTree.GetNodeLevel Method

[TBaseVirtualTree Class](#)

Returns the indentation level of the given node.

Pascal

```
function GetNodeLevel(Node: PVirtualNode): Cardinal;
```

Description

GetNodeLevel returns the level of **Node**. This level is determined by the number of parent nodes (excluding the hidden root node). Top level nodes have the level 0, their direct child nodes have level 1 etc.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetNodeLevel](#)
Method |
[TBaseVirtualTree.GetPrevious](#)
Method

TBaseVirtualTree.GetOptionsClass Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function GetOptionsClass: TTreeOptionsClass; virtual
```

Description

GetOptionsClass is a special purpose method to return a certain class which is used by the tree for its options. TVirtualBaseTree always returns [TCustomVirtualTreeOptions](#) 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(TCustomVirtualStringTr
private
    function GetOptions: TStringTreeOptions;
    procedure SetOptions(const Value: TStringTreeOp
protected
    function GetOptionsClass: TTreeOptionsClass; ov
public
    property Canvas;
published
    ...
    property TreeOptions: TStringTreeOptions read G
    ...
end;

...

//----- TVirtualStringTree -----

function TVirtualStringTree.GetOptions: TStringTree

begin
    Result := FOptions as TStringTreeOptions;
end;

//-----

procedure TVirtualStringTree.SetOptions(const Value

begin
    FOptions.Assign(Value);
end;

//-----

function TVirtualStringTree.GetOptionsClass: TTreeO

```

```
begin
  Result := TStringTreeOptions;
end;
```

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



TBaseVirtualTree Class |
TBaseVirtualTree.GetOptionsClass
Method |
TBaseVirtualTree.GetSortedCutCopySet
Method

TBaseVirtualTree.GetPrevious Method

TBaseVirtualTree Class

Group of node navigation functions.

Pascal

```
function GetPrevious(Node: PVirtualNode): PVirtualNo  
function GetPreviousInitialized(Node: PVirtualNode):  
function GetPreviousNoInit(Node: PVirtualNode): PVir  
function GetPreviousSibling(Node: PVirtualNode): PVi  
function GetPreviousVisible(Node: PVirtualNode): PVi  
function GetPreviousVisibleNoInit(Node: PVirtualNode  
function GetPreviousVisibleSibling(Node: PVirtualNod  
function GetPreviousVisibleSiblingNoInit(Node: PVirt
```

Description

This group of navigation functions is used to return the previous node relative to a given node in the tree with various properties.

GetPrevious	Previous node in the tree with initialization.
GetPreviousInitialized	Previous initialized node in the tree (no initialization needed).
GetPreviousNoInit	Previous node in the tree without initialization.
GetPreviousSibling	Previous sibling node with initialization.

GetPreviousVisible	Previous visible node in the tree with initialization.
GetPreviousVisibleNoInit	Previous visible node in the tree without initialization.
GetPreviousVisibleSibling	Previous visible sibling node with initialization.
GetPreviousVisibleSiblingNoInit	Previous visible sibling node without initialization.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetPrevious](#)
[Method](#) |
[TBaseVirtualTree.GetSortedSelection](#)
[Method](#)

TBaseVirtualTree.GetSortedCutCopySet Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function GetSortedCutCopySet(Resolve: Boolean): TNode;
```

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 Class](#) |
[TBaseVirtualTree.GetSortedCutCopySet
Method](#) | [TBaseVirtualTree.GetTextInfo
Method](#)

TBaseVirtualTree.GetSortedSelection Method

[TBaseVirtualTree Class](#)

Returns a sorted list of all currently selected nodes.

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](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetSortedSelection](#)
[Method](#) |
[TBaseVirtualTree.GetTreeFromDataObject](#)
[Method](#)

TBaseVirtualTree.GetTextInfo Method

[TBaseVirtualTree Class](#)

Helper method for node editors, hints etc.

Pascal

```
procedure GetTextInfo(Node: PVirtualNode; Column: TC
```

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 Class](#) |
[TBaseVirtualTree.GetTextInfo](#)
Method |
[TBaseVirtualTree.GetTreeRect](#)
Method

TBaseVirtualTree.GetTreeFromDataObject Method

[TBaseVirtualTree Class](#)

OLE drag'n drop and clipboard support method.

Pascal

```
function GetTreeFromDataObject(const DataObject: IDa
```

Description

Returns the owner/sender of the given data object by means of a special clipboard format or nil if the sender is in another process or no virtual tree at all.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetTreeFromDataObject](#)
[Method](#) |
[TBaseVirtualTree.GetVisibleParent](#)
[Method](#)

TBaseVirtualTree.GetTreeRect Method

[TBaseVirtualTree Class](#)

Returns the size of the virtual tree image.

Pascal

```
function GetTreeRect: TRect;
```

Description

GetTreeRect can be used to determine the full size of the [tree image](#) as used for painting etc.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetTreeRect](#)
[Method](#) |
[TBaseVirtualTree.HandleHotTrack](#)
[Method](#)

TBaseVirtualTree.GetVisibleParent Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function GetVisibleParent(Node: PVirtualNode): PVirt
```

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.GetVisibleParent Method](#) |
[TBaseVirtualTree.HandleIncrementalSearch Method](#)

TBaseVirtualTree.HandleHotTrack Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure HandleHotTrack(X: Integer; Y: Integer); vi
```

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 Class](#) |
[TBaseVirtualTree.HandleHotTrack](#)
[Method](#) |
[TBaseVirtualTree.HandleMouseDownClick](#)
[Method](#)

TBaseVirtualTree.HandleIncrementalSearch Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure HandleIncrementalSearch(CharCode: Word); v
```

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 Class](#) |
[TBaseVirtualTree.HandleIncrementalSearch](#)
[Method](#) |
[TBaseVirtualTree.HandleMouseDown](#)
[Method](#)

TBaseVirtualTree.HandleMouseDownDb1Click Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure HandleMouseDownDb1Click(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 Class](#) |
[TBaseVirtualTree.HandleMouseDownClick](#)
Method |
[TBaseVirtualTree.HandleMouseUp](#)
Method

TBaseVirtualTree.HandleMouseDown Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure HandleMouseDown(var Message: TWMMouse; con
```

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 Class](#) |
[TBaseVirtualTree.HandleMouseDown](#)
Method |
[TBaseVirtualTree.HasAsParent](#)
Method

TBaseVirtualTree.HandleMouseUp Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.HandleMouseUp](#)
[Method](#) |
[TBaseVirtualTree.HasImage](#)
[Method](#)

TBaseVirtualTree.HasAsParent Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function HasAsParent(Node: PVirtualNode; PotentialPa
```

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 Class](#) |
[TBaseVirtualTree.HasAsParent](#)
Method |
[TBaseVirtualTree.HasPopupMenu](#)
Method

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 Class](#) |
[TBaseVirtualTree.HasImage](#)
[Method](#) |
[TBaseVirtualTree.InitChildren](#)
[Method](#)

TBaseVirtualTree.HasPopupMenu Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function HasPopupMenu(Node: PVirtualNode; Column: TC
```

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 Class](#) |
[TBaseVirtualTree.HasPopupMenu](#)
[Method](#) |
[TBaseVirtualTree.InitNode](#) Method

TBaseVirtualTree.InitChildren Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure InitChildren(Node: PVirtualNode); virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.InitChildren](#)
Method |
[TBaseVirtualTree.InsertNode](#)
Method

TBaseVirtualTree.InitNode Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure InitNode(Node: PVirtualNode); virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.InitNode Method](#) |
[TBaseVirtualTree.InternalAddFromStream Method](#)

TBaseVirtualTree.InsertNode Method

[TBaseVirtualTree Class](#)

Inserts a new node and returns it to the caller.

Pascal

```
function InsertNode(Node: PVirtualNode; Mode: TVTNodeMode): PVirtualNode;
```

Description

Adds a new node relative to **Node**. The final position is determined by **Mode**. **UserData** can be used to set the first 4 bytes of the user data area to an initial value, which can be used in [OnInitNode](#) 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 validate 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 Class](#) |
[TBaseVirtualTree.InsertNode Method](#) |
[TBaseVirtualTree.InternalCacheNode Method](#)

TBaseVirtualTree.InternalAddFromStream Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure InternalAddFromStream(Stream: TStream; Ver
```

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 Class |
TBaseVirtualTree.InternalAddToSelection
Method (TNodeArray, Integer, Boolean)

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 Class](#) |
[TBaseVirtualTree.InternalAddToSelection
Method \(PVirtualNode, Boolean\)](#)

TBaseVirtualTree.InternalAddToSelection Method (TNodeArray, Integer, Boolean)

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function InternalAddToSelection(const NewItems: TNodeArray; Index: Integer; Selected: Boolean): 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 Class](#) |
[TBaseVirtualTree.InternalAddFromStream](#)
Method |
[TBaseVirtualTree.InternalClearSelection](#)
Method

TBaseVirtualTree.InternalCacheNode Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure InternalCacheNode(Node: PVirtualNode); var
```

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 Class](#) |
[TBaseVirtualTree.InternalCacheNode](#)
[Method](#) |
[TBaseVirtualTree.InternalConnectNode](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.InternalClearSelection
Method](#) | [TBaseVirtualTree.InternalData
Method](#)

TBaseVirtualTree.InternalConnectNode Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure InternalConnectNode(Node: PVirtualNode; De
```

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 Class](#) |
[TBaseVirtualTree.InternalConnectNode](#)
[Method](#) |
[TBaseVirtualTree.InternalDisconnectNode](#)
[Method](#)

TBaseVirtualTree.InternalData Method

[TBaseVirtualTree Class](#) | [See Also](#)

Returns the address of the internal data for a tree class.

Pascal

```
function InternalData(Node: PVirtualNode): Pointer;
```

Description

In TBaseVirtualTreeview this method returns nil but should be overridden in descendants to allow proper access to the internal data of **Node** if the descendant tree has allocated internal data.

See Also

[Data handling](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.InternalData Method](#) |
[TBaseVirtualTree.InternalRemoveFromSelection Method](#)

TBaseVirtualTree.InternalDisconnectNode Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure InternalDisconnectNode(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 Class](#) |
[TBaseVirtualTree.InternalDisconnectNode](#)
Method |
[TBaseVirtualTree.InvalidateCache](#)
Method

TBaseVirtualTree.InternalRemoveFromSe 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 Class](#) |
[TBaseVirtualTree.InternalRemoveFromSelection
Method](#) | [TBaseVirtualTree.InvalidateChildren
Method](#)

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 Class](#) |
[TBaseVirtualTree.InvalidateCache](#)
Method |
[TBaseVirtualTree.InvalidateColumn](#)
Method

TBaseVirtualTree.InvalidateChildren Method

[TBaseVirtualTree Class](#)

Invalidates all children of the given node.

Pascal

```
procedure InvalidateChildren(Node: PVirtualNode; Rec
```

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 Class](#) |
[TBaseVirtualTree.InvalidateChildren](#)
Method |
[TBaseVirtualTree.InvalidateNode](#)
Method

TBaseVirtualTree.InvalidateColumn Method

[TBaseVirtualTree Class](#)

Invalidates the client area part of a column.

Pascal

```
procedure InvalidateColumn(Column: TColumnIndex);
```

Description

Invalidates the client area part of a column.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.InvalidateColumn](#)
Method |
[TBaseVirtualTree.InvalidateToBottom](#)
Method

TBaseVirtualTree.InvalidateNode Method

[TBaseVirtualTree Class](#)

Invalidates the given node.

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 Class](#) |
[TBaseVirtualTree.InvalidateNode](#)
Method |
[TBaseVirtualTree.InvertSelection](#)
Method

TBaseVirtualTree.InvalidateToBottom Method

[TBaseVirtualTree Class](#)

Invalidates the client area starting with the top position of the given node.

Pascal

```
procedure InvalidateToBottom(Node: PVirtualNode);
```

Description

InvalidateToBottom initiates repaint of client area starting at given node. If this node is not visible or not yet initialized then nothing happens.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.InvalidateToBottom
Method](#) | [TBaseVirtualTree.IsEditing
Method](#)

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 Class](#) |
[TBaseVirtualTree.InvertSelection](#)
[Method](#) |
[TBaseVirtualTree.IsMouseSelecting](#)
[Method](#)

TBaseVirtualTree.IsEditing Method

[TBaseVirtualTree Class](#)

Tells the caller whether the tree is currently in edit mode.

Pascal

```
function IsEditing: Boolean;
```

Description

Just a simple shortcut to test the tsEditing state.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.IsEditing Method](#)
| [TBaseVirtualTree.IterateSubtree Method](#)

TBaseVirtualTree.IsMouseSelecting Method

[TBaseVirtualTree Class](#)

Tell the caller whether the tree is currently in draw selection mode.

Pascal

```
function IsMouseSelecting: Boolean;
```

Description

IsMouseSelecting returns true if draw selection by the user is active or pending.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.IsMouseSelecting
Method](#) | [TBaseVirtualTree.Loaded
Method](#)

TBaseVirtualTree.IterateSubtree Method

[TBaseVirtualTree Class](#)

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.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.IterateSubtree](#)
Method |
[TBaseVirtualTree.LoadFromFile](#)
Method

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 Class](#) |
[TBaseVirtualTree.Loaded Method](#) |
[TBaseVirtualTree.MainColumnChanged Method](#)

TBaseVirtualTree.LoadFromFile Method

[TBaseVirtualTree Class](#) | [See Also](#)

Loads previously streamed out tree data back in again.

Pascal

```
procedure LoadFromFile(const FileName: TFileName); v  
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 Class](#) |
[TBaseVirtualTree.LoadFromFile](#)
[Method](#) |
[TBaseVirtualTree.MarkCutCopyNodes](#)
[Method](#)

TBaseVirtualTree.MainColumnChanged Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure MainColumnChanged; virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.MainColumnChanged](#)
[Method](#) |
[TBaseVirtualTree.MeasureItemHeight](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.MarkCutCopyNodes](#)
Method |
[TBaseVirtualTree.MouseMove](#)
Method

TBaseVirtualTree.MeasureItemHeight Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.MeasureItemHeight](#)
[Method](#) |
[TBaseVirtualTree.Notification Method](#)

TBaseVirtualTree.MouseMove Method

[TBaseVirtualTree Class](#)

Not documented.

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)

TBaseVirtualTree Class

Moves **Source** and all its child nodes to **Target**.

Pascal

```
procedure MoveTo(Source: PVirtualNode; Target: PVirt  
procedure MoveTo(Node: PVirtualNode; Tree: TBaseVirt
```

Description

Moves the given node (and all its children) to **Target**. **Source** must belong to the tree instance which calls this MoveTo method. **Mode** determines how to connect **Source** to **Target**. This method might involve a change of the tree if **Target** belongs to a different tree than **Source**.

The variant taking a tree reference as target can be used to transfer nodes to a different tree, without determining its root node first. However one can also pass in any virtual tree node as target, as long as it belongs to a tree. The owning tree is automatically determined and an optimized path is taken if the operation happens within one tree. In this case simply the source node is disconnected from the old place and reconnected at the new location.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.MouseMove](#)
[Method](#) |
[TBaseVirtualTree.OriginalWMNCPaint](#)
[Method](#)

TBaseVirtualTree.Notification Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure Notification(AComponent: TComponent; Opera
```

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 Class](#) |
[TBaseVirtualTree.Notification
Method](#) | [TBaseVirtualTree.Paint
Method](#)

TBaseVirtualTree.OriginalWMNCPaint Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure OriginalWMNCPaint(DC: HDC); virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.OriginalWMNCPaint](#)
Method |
[TBaseVirtualTree.PaintCheckImage](#)
Method

TBaseVirtualTree.Paint Method

[TBaseVirtualTree Class](#)

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

Pascal

```
procedure Paint; override;
```

Description

Overriden method to paint the tree image. The actual work is however done in [PaintTree](#).

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.Paint Method](#) |
[TBaseVirtualTree.PaintImage Method](#)

TBaseVirtualTree.PaintCheckImage Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure PaintCheckImage(const PaintInfo: TVTPaintI
```

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 Class](#) |
[TBaseVirtualTree.PaintCheckImage](#)
Method |
[TBaseVirtualTree.PaintNodeButton](#)
Method

TBaseVirtualTree.PaintImage Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure PaintImage(var PaintInfo: TVTPaintInfo; Im
```

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 Class](#) |
[TBaseVirtualTree.PaintImage Method](#) |
[TBaseVirtualTree.PaintSelectionRectangle Method](#)

TBaseVirtualTree.PaintNodeButton Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure PaintNodeButton(Canvas: TCanvas; Node: PVi
```

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 Class](#) |
[TBaseVirtualTree.PaintNodeButton](#)
[Method](#) |
[TBaseVirtualTree.PaintTree](#)
[Method](#)

TBaseVirtualTree.PaintSelectionRectangle Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure PaintSelectionRectangle(Target: TCanvas; W
```

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 Class](#) |
[TBaseVirtualTree.PaintSelectionRectangle
Method](#) | [TBaseVirtualTree.PaintTreeLines
Method](#)

TBaseVirtualTree.PaintTree Method

[TBaseVirtualTree Class](#) | [See Also](#)

Main paint routine for the tree image.

Pascal

```
procedure PaintTree(TargetCanvas: TCanvas; Window: T
```

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 Class](#) |
[TBaseVirtualTree.PaintTree Method](#) |
[TBaseVirtualTree.PanningWindowProc Method](#)

TBaseVirtualTree.PaintTreeLines Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure PaintTreeLines(const PaintInfo: TVTPaintIn
```

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 Class](#) |
[TBaseVirtualTree.PaintTreeLines](#)
Method |
[TBaseVirtualTree.PasteFromClipboard](#)
Method

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 Class](#) |
[TBaseVirtualTree.PanningWindowProc](#)
[Method](#) |
[TBaseVirtualTree.PrepareDragImage](#)
[Method](#)

TBaseVirtualTree.PasteFromClipboard Method

[TBaseVirtualTree Class](#)

Inserts the content of the clipboard into the tree.

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 Class](#) |
[TBaseVirtualTree.PasteFromClipboard](#)
[Method](#) | [TBaseVirtualTree.Print](#)
[Method](#)

TBaseVirtualTree.PrepareDragImage Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure PrepareDragImage(HotSpot: TPoint; const Da
```

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 Class](#) |
[TBaseVirtualTree.PrepareDragImage](#)
Method |
[TBaseVirtualTree.ProcessDrop](#)
Method

TBaseVirtualTree.Print Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure Print(Printer: TPrinter; PrintHeader: Bool
```

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 Class](#) |
[TBaseVirtualTree.Print Method](#) |
[TBaseVirtualTree.ProcessOLEData Method](#)

TBaseVirtualTree.ProcessDrop Method

[TBaseVirtualTree Class](#)

Helper method to ease OLE drag'n drop operations.

Pascal

```
function ProcessDrop(DataObject: IDataObject; Target
```

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 Class](#) |
[TBaseVirtualTree.ProcessDrop](#)
Method |
[TBaseVirtualTree.ReadChunk](#)
Method

TBaseVirtualTree.ProcessOLEData Method

[TBaseVirtualTree Class](#)

Takes serialized OLE tree data and reconstructs the former structure.

Pascal

```
function ProcessOLEData(Source: TBaseVirtualTree; Da
```

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.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ProcessOLEData](#)
[Method](#) |
[TBaseVirtualTree.ReadNode](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.ReadChunk Method](#) |
[TBaseVirtualTree.RedirectFontChangeEvent Method](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ReadNode](#)
Method |
[TBaseVirtualTree.ReinitChildren](#)
Method

TBaseVirtualTree.RedirectFontChangeEvent Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.RedirectFontChangeEvent
Method](#) | [TBaseVirtualTree.ReinitNode
Method](#)

TBaseVirtualTree.ReinitChildren Method

[TBaseVirtualTree Class](#)

Forces all child nodes of Node to be reinitialized.

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ReinitChildren Method](#) |
[TBaseVirtualTree.RemoveFromSelection Method](#)

TBaseVirtualTree.ReinitNode Method

[TBaseVirtualTree Class](#)

Forces a reinitialization of the given node.

Pascal

```
procedure ReinitNode(Node: PVirtualNode; Recursive:
```

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 Class](#) |
[TBaseVirtualTree.ReinitNode](#)
Method |
[TBaseVirtualTree.RenderOLEData](#)
Method

TBaseVirtualTree.RemoveFromSelection Method

[TBaseVirtualTree Class](#)

Removes the given node from the current selection.

Pascal

```
procedure RemoveFromSelection(Node: PVirtualNode); v
```

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 Class](#) |
[TBaseVirtualTree.RemoveFromSelection
Method](#) | [TBaseVirtualTree.RepaintNode
Method](#)

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 Class](#) |
[TBaseVirtualTree.RenderOLEData](#)
[Method](#) |
[TBaseVirtualTree.ResetNode](#)
[Method](#)

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 Class](#) |
[TBaseVirtualTree.RepaintNode](#)
[Method](#) |
[TBaseVirtualTree.ResetRangeAnchor](#)
[Method](#)

TBaseVirtualTree.ResetNode Method

[TBaseVirtualTree Class](#)

Resets the given node to uninitialized.

Pascal

```
procedure ResetNode(Node: PVirtualNode); virtual;
```

Description

ResetNode deletes all children of **Node** and marks it as being uninitialized.

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ResetNode Method](#) |
[TBaseVirtualTree.RestoreFontChangeEvent Method](#)

TBaseVirtualTree.ResetRangeAnchor Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure ResetRangeAnchor; virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ResetRangeAnchor](#)
[Method](#) |
[TBaseVirtualTree.SaveToFile Method](#)

TBaseVirtualTree.RestoreFontChangeEvent Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure RestoreFontChangeEvent(Canvas: TCanvas); v
```

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 Class](#) |
[TBaseVirtualTree.RestoreFontChangeEvent
Method](#) | [TBaseVirtualTree.ScrollIntoView
Method](#)

TBaseVirtualTree.SaveToFile Method

[TBaseVirtualTree Class](#) | [See Also](#)

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

Pascal

```
procedure SaveToFile(const FileName: TFileName);  
procedure SaveToStream(Stream: TStream; Node: PVirtu
```

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 Class](#) |
[TBaseVirtualTree.SaveToFile](#)
[Method](#) |
[TBaseVirtualTree.SelectAll](#) Method

TBaseVirtualTree.ScrollIntoView Method

[TBaseVirtualTree Class](#)

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

Pascal

```
function ScrollIntoView(Node: PVirtualNode; Center:
```

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 Class](#) |
[TBaseVirtualTree.ScrollIntoView](#)
[Method](#) |
[TBaseVirtualTree.SelectNodes](#)
[Method](#)

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

```
procedure SelectNodes(StartNode: PVirtualNode; EndNo
```

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 Class](#) |
[TBaseVirtualTree.SelectNodes Method](#) |
[TBaseVirtualTree.SetFocusedNodeAndColumn Method](#)

TBaseVirtualTree.SetBiDiMode Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure SetBiDiMode(Value: TBiDiMode); override;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.SetBiDiMode](#)
Method |
[TBaseVirtualTree.SkipNode](#)
Method

TBaseVirtualTree.SetFocusedNodeAndCo Method

[TBaseVirtualTree Class](#)

Not documented.

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 Class](#) |
[TBaseVirtualTree.SetFocusedNodeAndColumn
Method](#) | [TBaseVirtualTree.Sort Method](#)

TBaseVirtualTree.SkipNode Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure SkipNode(Stream: TStream); virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.SkipNode](#)
[Method](#) |
[TBaseVirtualTree.SortTree Method](#)

TBaseVirtualTree.Sort Method

[TBaseVirtualTree Class](#)

Sorts the given node.

Pascal

```
procedure Sort(Node: PVirtualNode; Column: TColumnIn
```

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 Class](#) |
[TBaseVirtualTree.Sort Method](#) |
[TBaseVirtualTree.StartWheelPanning Method](#)

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 Class](#) |
[TBaseVirtualTree.SortTree Method](#) |
[TBaseVirtualTree.StopWheelPanning Method](#)

TBaseVirtualTree.StartWheelPanning Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure StartWheelPanning(Position: TPoint); virtu
```

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 Class](#) |
[TBaseVirtualTree.StartWheelPanning](#)
Method |
[TBaseVirtualTree.StructureChange](#)
Method

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 Class](#) |
[TBaseVirtualTree.StopWheelPanning](#)
Method |
[TBaseVirtualTree.SuggestDropEffect](#)
Method

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 Class](#) |
[TBaseVirtualTree.StructureChange](#)
[Method](#) |
[TBaseVirtualTree.ToggleNode](#)
[Method](#)

TBaseVirtualTree.SuggestDropEffect Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
function SuggestDropEffect(Source: TObject; Shift: 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 Class](#) |
[TBaseVirtualTree.SuggestDropEffect](#)
Method |
[TBaseVirtualTree.ToggleSelection](#)
Method

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 Class](#) |
[TBaseVirtualTree.ToggleNode](#)
[Method](#) |
[TBaseVirtualTree.UnselectNodes](#)
[Method](#)

TBaseVirtualTree.ToggleSelection Method

[TBaseVirtualTree Class](#)

Toggles the selection state of a range of nodes.

Pascal

```
procedure ToggleSelection(StartNode: PVirtualNode; E
```

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ToggleSelection](#)
Method |
[TBaseVirtualTree.UpdateAction](#)
Method

TBaseVirtualTree.UnselectNodes Method

[TBaseVirtualTree Class](#)

Deselects a range of nodes.

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 Class](#) |
[TBaseVirtualTree.UnselectNodes](#)
Method |
[TBaseVirtualTree.UpdateDesigner](#)
Method

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 Class](#) |
[TBaseVirtualTree.UpdateAction](#)
[Method](#) |
[TBaseVirtualTree.UpdateEditBounds](#)
[Method](#)

TBaseVirtualTree.UpdateDesigner Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure UpdateDesigner; virtual;
```

Description

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

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.UpdateDesigner](#)
[Method](#) |
[TBaseVirtualTree.UpdateHeaderRect](#)
[Method](#)

TBaseVirtualTree.UpdateEditBounds Method

[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 Class](#) |
[TBaseVirtualTree.UpdateEditBounds](#)
Method |
[TBaseVirtualTree.UpdateScrollBars](#)
Method

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 Class](#) |
[TBaseVirtualTree.UpdateHeaderRect Method](#) |
[TBaseVirtualTree.UpdateWindowAndDragImage Method](#)

TBaseVirtualTree.UpdateScrollBars Method

[TBaseVirtualTree Class](#)

Applies changes to the horizontal and vertical scrollbars.

Pascal

```
procedure UpdateHorizontalScrollBar(DoRepaint: Boolean);  
procedure UpdateScrollBars(DoRepaint: Boolean); virt  
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 Class](#) |
[TBaseVirtualTree.UpdateScrollBars](#)
[Method](#) |
[TBaseVirtualTree.UseRightToLeftReading](#)
[Method](#)

TBaseVirtualTree.UpdateWindowAndDrag Method

[TBaseVirtualTree Class](#)

Not documented.

Pascal

```
procedure UpdateWindowAndDragImage(const Tree: TBase
```

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 Class](#) |
[TBaseVirtualTree.UpdateWindowAndDragImage
Method](#) | [TBaseVirtualTree.ValidateCache
Method](#)

TBaseVirtualTree.UseRightToLeftReading Method

[TBaseVirtualTree Class](#)

Helper method for right-to-left layout.

Pascal

```
function UseRightToLeftReading: Boolean;
```

Description

UseRightToLeftReading had to be overridden 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 Class](#) |
[TBaseVirtualTree.UseRightToLeftReading](#)
[Method](#) |
[TBaseVirtualTree.ValidateChildren](#)
[Method](#)

TBaseVirtualTree.ValidateCache Method

[TBaseVirtualTree Class](#) | [See Also](#)

Initiates the validation of the internal node cache.

Pascal

```
procedure ValidateCache; virtual;
```

Description

If the node cache is marked as being invalid then this method puts the tree into the worker thread's list and awakes then the thread so that the validation is performed in the background.

See Also

[InvalidateCache](#)

Class

[TBaseVirtualTree Class](#)

Links

[TBaseVirtualTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ValidateCache](#)
Method |
[TBaseVirtualTree.ValidateNode](#)
Method

TBaseVirtualTree.ValidateChildren Method

[TBaseVirtualTree Class](#)

Validates all children of a given node.

Pascal

```
procedure ValidateChildren(Node: PVirtualNode; Recur
```

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](#)

What do you think about this topic? [Send feedback!](#)



[TBaseVirtualTree Class](#) |
[TBaseVirtualTree.ValidateChildren](#)
Method |
[TBaseVirtualTree.ValidateNodeDataSize](#)
Method

TBaseVirtualTree.ValidateNode Method

[TBaseVirtualTree Class](#)

Validates a given node.

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 Class](#) |
[TBaseVirtualTree.ValidateNode](#)
[Method](#) |
[TBaseVirtualTree.WndProc](#)
[Method](#)

TBaseVirtualTree.ValidateNodeDataSize Method

[TBaseVirtualTree Class](#)

Helper method for node data size initialization.

Pascal

```
procedure ValidateNodeDataSize(var Size: Integer); v
```

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 Class](#) |
[TBaseVirtualTree.ValidateNodeDataSize
Method](#) | [TBaseVirtualTree.WriteChunks
Method](#)

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 Class](#) |
[TBaseVirtualTree.WndProc](#)
[Method](#) |
[TBaseVirtualTree.WriteNode](#)
[Method](#)

TBaseVirtualTree.WriteChunks Method

[TBaseVirtualTree Class](#) | [See Also](#)

Writes the core chunks for the given node to the given stream.

Pascal

```
procedure WriteChunks(Stream: TStream; Node: PVirtualTree);
```

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 Class](#) |
[TBaseVirtualTree.WriteChunks](#)
Method

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: PVirtualIN
```

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!](#)



[Classes](#) | [TBaseVirtualTree Class](#) |
[TClipboardFormatList Class](#)

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

`TBufferedString`

File

VirtualTrees

Links

[Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TBufferedString.AsString Property

[TBufferedString Class](#)

Not documented.

Pascal

```
property AsString: string;
```

Description

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

Class

[TBufferedString Class](#)

Links

[TBufferedString Class](#)

What do you think about this topic? [Send feedback!](#)



[TBufferedString Class](#) |
[TBufferedString.AsString Property](#)
| [TBufferedString.AddNewLine](#)
Method

TBufferedString.Add Method

[TBufferedString Class](#)

Not documented.

Pascal

```
procedure Add(const S: string);
```

Description

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

Class

[TBufferedString Class](#)

Links

[TBufferedString Class](#)

What do you think about this topic? [Send feedback!](#)



[TBufferedString Class](#) |
[TBufferedString.Add Method](#) |
[TBufferedString.Destroy](#)
[Destructor](#)

TBufferedString.AddNewLine Method

[TBufferedString Class](#)

Not documented.

Pascal

```
procedure AddNewLine;
```

Description

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

Class

[TBufferedString Class](#)

Links

[TBufferedString Class](#)

What do you think about this topic? [Send feedback!](#)



TBufferedString.Destroy Destructor

TBufferedString 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

[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

TClipboardFormatList

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

```
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 Class](#) |
[TClipboardFormatList.Add Method](#)
| [TClipboardFormatList.Create
Constructor](#)

TClipboardFormatList.Clear Method

[TClipboardFormatList Class](#)

Not documented.

Pascal

```
procedure Clear;
```

Description

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

Class

[TClipboardFormatList Class](#)

Links

[TClipboardFormatList Class](#)

What do you think about this topic? [Send feedback!](#)



[TClipboardFormatList Class](#) |
[TClipboardFormatList.Clear](#)
Method |
[TClipboardFormatList.Destroy](#)
Destructor

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.Destroy 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 Class](#) |
[TClipboardFormatList.EnumerateFormats](#)
Method (TVirtualTreeClass, TStrings)

TClipboardFormatList.EnumerateFormats Method (TVirtualTreeClass, TFormatEtcArray, TClipboardFormats)

[TClipboardFormatList Class](#)

Returns a list of format records for the given class.

Pascal

```
procedure EnumerateFormats(TreeClass: TVirtualTreeCl
```

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 ([TVirtualTreeClass](#),
[TFormatEtcArray](#), [TClipboardFormats](#))

TClipboardFormatList.EnumerateFormats Method (TVirtualTreeClass, TStrings)

[TClipboardFormatList Class](#)

Returns a list of format descriptions for the given class.

Pascal

```
procedure EnumerateFormats(TreeClass: TVirtualTreeCl
```

Class

[TClipboardFormatList Class](#)

Links

[TClipboardFormatList Class](#)

What do you think about this topic? [Send feedback!](#)



TClipboardFormatList Class |
TClipboardFormatList.FindFormat
Method (string)

TClipboardFormatList.FindFormat Method (Word, string)

TClipboardFormatList Class

Not documented.

Pascal

```
function FindFormat(Fmt: Word; var Description: string): 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 Class](#) |
[TClipboardFormatList.FindFormat
Method \(Word, string\)](#) |
[TClipboardFormatList.FindFormat
Method \(string, Word\)](#)

TClipboardFormatList.FindFormat Method (string)

[TClipboardFormatList Class](#)

Not documented.

Pascal

```
function FindFormat(FormatString: string): PClipboard
```

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 Class |
TClipboardFormatList.FindFormat
Method (string)

TClipboardFormatList.FindFormat Method (string, Word)

TClipboardFormatList Class

Not documented.

Pascal

```
function FindFormat(FormatString: string; var Fmt: W
```

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!](#)



[Classes](#) | [TClipboardFormatList Class](#) | [TCriticalSection Class](#)

TClipboardFormats Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

List of strings describing clipboard formats.

Pascal

```
TClipboardFormats = class(TStringList);
```

Description

This class is an extended string list which allows to enter description strings for clipboard formats which are checked 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 Class](#) |
[TClipboardFormats.Add Method](#)

TClipboardFormats.Owner Property

[TClipboardFormats Class](#)

Not documented.

Pascal

```
property Owner: TBaseVirtualTree;
```

Description

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

Class

[TClipboardFormats Class](#)

Links

[TClipboardFormats Class](#)

What do you think about this topic? [Send feedback!](#)



[TClipboardFormats Class](#) |
[TClipboardFormats.Owner](#)
[Property](#) |
[TClipboardFormats.Create](#)
[Constructor](#)

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); o
```

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 Class |
TClipboardFormats.Add Method

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

 [FSection](#)

Not documented.

Methods

 [Create](#)

Not documented.

 [Destroy](#)

Not documented.

 [Enter](#)

Not documented.

 [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 Class](#) |
[TCriticalSection.Create](#)
Constructor

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 Class](#) |
[TCriticalSection.FSection Field](#) |
[TCriticalSection.Destroy](#)
Destructor

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 Class](#) |
[TCriticalSection.Create](#)
[Constructor](#) |
[TCriticalSection.Enter Method](#)

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 Class](#) |
[TCriticalSection.Destroy](#)
[Destructor](#) | [TCriticalSection.Leave](#)
[Method](#)

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 Class](#) |
[TCriticalSection.Enter Method](#)

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!](#)



[Classes](#) | [TCriticalSection Class](#) | [TCustomVirtualDrawTree Class](#)

TCustomStringTreeOptions Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Enhanced options class for string trees.

Pascal

```
TCustomStringTreeOptions = class(TCustomVirtualTreeO
```

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

Links

[Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TCustomStringTreeOptions.StringOptions Property

[TCustomStringTreeOptions Class](#)

The new options introduced by the class.

Pascal

```
property StringOptions: TVTStringOptions;
```

Description

StringOptions provides access to the newly introduced options by which the base class is extended.

Class

[TCustomStringTreeOptions Class](#)

Links

[TCustomStringTreeOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomStringTreeOptions Class](#) |
[TCustomStringTreeOptions.StringOptions](#)
[Property](#) |
[TCustomStringTreeOptions.Create](#)
[Constructor](#)

TCustomStringTreeOptions.AssignTo Method

[TCustomStringTreeOptions Class](#)

Used to copy the options class.

Pascal

```
procedure AssignTo(Dest: TPersistent); override;
```

Description

You can either call this method directly or use the Assign method of the target class to do the assignment. Implementing AssignTo instead of Assign allows for future enhancements. TPersistent will automatically call AssignTo if there was no Assign method.

Class

[TCustomStringTreeOptions Class](#)

Links

[TCustomStringTreeOptions Class](#)

What do you think about this topic? [Send feedback!](#)



TCustomStringTreeOptions.Create Constructor

TCustomStringTreeOptions Class

The constructor of the class.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree); overri
```

Description

The constructor initializes the class.

Class

TCustomStringTreeOptions Class

Links

TCustomStringTreeOptions Class

What do you think about this topic? [Send feedback!](#)



[Classes](#) |
[TCustomStringTreeOptions Class](#) |
[TCustomVirtualStringTree Class](#)

TCustomVirtualDrawTree Class

[Events](#) | [Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Simple owner draw descendant of the base tree.

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.
- ➤ **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.

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**

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.

   **ColumnsIsEmpty**

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 OnGetPopupMenu 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.
- 🟢🔗 **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 s 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.
-  **HandleMouseDownClick**
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



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

```
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](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualDrawTree Class](#) |
[TCustomVirtualDrawTree.OnDrawHint](#)
[Event](#) |
[TCustomVirtualDrawTree.OnGetHintSize](#)
[Event](#)

TCustomVirtualDrawTree.OnDrawNode Event

[TCustomVirtualDrawTree 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

[TCustomVirtualDrawTree Class](#)

Links

[TCustomVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualDrawTree Class](#) |
[TCustomVirtualDrawTree.OnDrawNode](#)
[Event](#) |
[TCustomVirtualDrawTree.OnGetNodeWidth](#)
[Event](#)

TCustomVirtualDrawTree.OnGetHintSize Event

[TCustomVirtualDrawTree 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

[TCustomVirtualDrawTree Class](#)

Links

[TCustomVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualDrawTree Class](#) |
[TCustomVirtualDrawTree.OnGetHintSize](#)
[Event](#) |
[TCustomVirtualDrawTree.DoDrawHint](#)
[Method](#)

TCustomVirtualDrawTree.OnGetNodeWidth Event

[TCustomVirtualDrawTree Class](#)

Triggered when a node is about to be drawn.

Pascal

```
property OnGetNodeWidth: TVTGetNodeWidthEvent;
```

Description

Use an event handler for OnGetNodeWidth to return your calculated width for the given node. Since the draw does not know the width of a node you have to tell it yourself.

Class

[TCustomVirtualDrawTree Class](#)

Links

[TCustomVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualDrawTree Class](#) |
[TCustomVirtualDrawTree.OnGetNodeWidth](#)
[Event](#) |
[TCustomVirtualDrawTree.DoGetHintSize](#)
[Method](#)

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 Class](#) |
[TCustomVirtualDrawTree.DoDrawHint](#)
Method |
[TCustomVirtualDrawTree.DoGetNodeWidth](#)
Method

TCustomVirtualDrawTree.DoGetHintSize Method

[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 Class](#) |
[TCustomVirtualDrawTree.DoGetHintSize](#)
Method |
[TCustomVirtualDrawTree.DoPaintNode](#)
Method

TCustomVirtualDrawTree.DoGetNodeWidth Method

[TCustomVirtualDrawTree Class](#)

Overridable method which triggers [OnGetNodeWidth](#).

Pascal

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

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

[TCustomVirtualDrawTree Class](#)

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!](#)



[Classes](#) |
[TCustomVirtualDrawTree Class](#) |
[TCustomVirtualTreeOptions Class](#)

TCustomVirtualStringTree Class

[Events](#) | [Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Descendant of [TBaseVirtualTree](#), which is able to manage node captions on its own

Pascal

```
TCustomVirtualStringTree = class(TBaseVirtualTree);
```

Description

TCustomVirtualStringTree enhances the base tree to display and edit node captions. It implements a generic node editor which can be used as reference to build your own one.

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.

➤ **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.

Methods

-    **AdjustPaintCellRect**
Method which can be used by descendants to adjust the given rectangle during a paint cycle.
-    **CalculateTextWidth**
Not documented.
-    **ColumnsIsEmpty**
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**

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.

  **ColumnsIsEmpty**

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 OnGetPopupMenu 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.
- 🟢🔗 **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 s 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.

 **HandleMouseDown**

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



File

VirtualTrees

Links

[Events](#), [Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TCustomVirtualStringTree Class |
TCustomVirtualStringTree.EllipsisWidth
Property

TCustomVirtualStringTree.DefaultText Property

TCustomVirtualStringTree Class

Not documented.

Pascal

```
property DefaultText: WideString;
```

Description

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

Class

TCustomVirtualStringTree Class

Links

TCustomVirtualStringTree Class

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.DefaultText](#)
Property |
[TCustomVirtualStringTree.OnGetHint](#)
Event

TCustomVirtualStringTree.EllipsisWidth Property

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
property EllipsisWidth: Integer;
```

Description

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

Class

[TCustomVirtualStringTree Class](#)

Links

[TCustomVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.EllipsisWidth](#)
Property |
[TCustomVirtualStringTree.OnGetText](#)
Event

TCustomVirtualStringTree.OnGetHint Event

[TCustomVirtualStringTree 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

[TCustomVirtualStringTree Class](#)

Links

[TCustomVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.OnGetHint](#)
[Event](#) |
[TCustomVirtualStringTree.OnNewText](#)
[Event](#)

TCustomVirtualStringTree.OnGetText Event

[TCustomVirtualStringTree Class](#) | [See Also](#)

Virtual string tree event to query for a node's normal or static text.

Pascal

```
property OnGetText: TVSTGetTextEvent;
```

Description

This is one of the fundamental string tree events which must always be handled. The string tree will fire this event every time when it needs to know about the text of a specific node and column. This is mainly the case when the node appears in the visible area of the tree view (in other words it is not scrolled out of view) but also on some other occasions, including streaming, drag and drop and calculating the width of the node.

The node text is distinguished between two text types:

- Normal text: If `TextType` is `ttNormal` return the main node caption

for the specified column.

- Static text: All text that you return when `TextType` is `ttStatic` will be displayed right beside the normal text (or left to it if the column's `BidiMode` is not `bdLeftToRight`, i.e. the column has right-to-left layout). Static text is used only for informational purposes; it cannot be selected or dragged and if the column is not wide enough to show all text it will not be shortened with an ellipsis (...) as normal text. The string tree will only query for static text if the `StringOptions` (see [TreeOptions](#)) include `ToShowStaticText`. This is off by default.

When this event is fired the text parameter will always be initialized with the value of property `DefaultText`. To handle the event get your node data and then extract the string for the appropriate column and `TextType`.

Notes

Be sure that your event handler only contains absolutely necessary code. This event will be fired very often - easily a

few hundred times for medium sized trees with some columns defined when the tree is repainted completely.

For example it is far too slow to use `Locate()` on some `Dataset`, a database query result or table, and then get the text

from some `TField`. This may only work with in-memory tables or a client dataset. When you initialize your node data do

some caching and use these cached values to display the data.

See Also

[OnPaintText](#)

Class

[TCustomVirtualStringTree Class](#)

Links

[TCustomVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.OnGetText](#)
[Event](#) |
[TCustomVirtualStringTree.OnPaintText](#)
[Event](#)

TCustomVirtualStringTree.OnNewText Event

[TCustomVirtualStringTree Class](#) | [See Also](#)

Virtual string tree event to pass edited text.

Pascal

```
property OnNewText: TVSTNewTextEvent;
```

Description

A string tree will fire this event after a node has been edited successfully (not canceled with Escape). The event handler must store the new text in the node data.

This event will only be used for the default node caption editor. Other custom node editors may or may not use this event to pass their edited data to the application. Editing for the whole tree is only possible if the MiscOptions (see [TreeOptions](#)) include toEditable. If only certain columns or nodes should be editable write an event handler for [OnEditing](#).

See Also

[OnCreateEditor](#), [OnEdited](#)

Class

TCustomVirtualStringTree Class

Links

[TCustomVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.OnNewText](#)
[Event](#) |
[TCustomVirtualStringTree.OnShortenString](#)
[Event](#)

TCustomVirtualStringTree.OnPaintText Event

[TCustomVirtualStringTree Class](#) | [See Also](#)

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 Class](#) |
[TCustomVirtualStringTree.OnPaintText
Event](#) | [TCustomVirtualStringTree.Text
Property](#)

TCustomVirtualStringTree.OnShortenString Event

[TCustomVirtualStringTree Class](#)

String tree event for custom handling of string abbreviations.

Pascal

```
property OnShortenString: TVSTShortenStringEvent;
```

Description

If the text of a node does not fit into its cell (in grid mode) or is too wide for the width of the tree view it is being abbreviated with an ellipsis (...). By default the ellipsis is added to the end of the node text.

Occasionally you may want to shorten the node text at a different position, for example if the node text is a **path** string and not the last folder or filename should be cut off but rather some mid level folders if possible.

In the handler S must be processed (shortened) and returned in Result. If Done is set to true (default value is false) the tree view takes over the shortening. This is useful if not all nodes or columns need

Class

[TCustomVirtualStringTree Class](#)

Links

TCustomVirtualStringTree Class

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.OnShortenString](#)
[Event](#) |
[TCustomVirtualStringTree.TreeOptions](#)
[Property](#)

TCustomVirtualStringTree.Text Property

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
property Text [Node: PVirtualNode; Column: TColumnIn
```

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 Class](#) |
[TCustomVirtualStringTree.Text Property](#) |
[TCustomVirtualStringTree.AdjustPaintCellRect](#)
Method

TCustomVirtualStringTree.TreeOptions Property

[TCustomVirtualStringTree Class](#)

Reference to the tree's options.

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 Class](#) |
[TCustomVirtualStringTree.TreeOptions](#)
[Property](#) |
[TCustomVirtualStringTree.CalculateTextWidth](#)
[Method](#)

TCustomVirtualStringTree.AdjustPaintCell Method

[TCustomVirtualStringTree Class](#)

Method which can be used by descendants to adjust the given rectangle during a paint cycle.

Pascal

```
procedure AdjustPaintCellRect(var PaintInfo: TVTPain
```

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 Class](#) |
[TCustomVirtualStringTree.AdjustPaintCellRect](#)
Method |
[TCustomVirtualStringTree.ColumnsIsEmpty](#)
Method

TCustomVirtualStringTree.CalculateTextWidth Method

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
function CalculateTextWidth(Canvas: TCanvas; Node: P
```

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 Class](#) |
[TCustomVirtualStringTree.CalculateTextWidth](#)
Method |
[TCustomVirtualStringTree.ComputeNodeHeight](#)
Method

TCustomVirtualStringTree.ColumnsIsEmpty Method

[TCustomVirtualStringTree Class](#) | [See Also](#)

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

Pascal

```
function ColumnsIsEmpty(Node: PVirtualNode; Column: T
```

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 Class](#) |
[TCustomVirtualStringTree.ColumnsIsEmpty](#)
Method |
[TCustomVirtualStringTree.ContentToClipboard](#)
Method

TCustomVirtualStringTree.ComputeNodeHeight Method

[TCustomVirtualStringTree Class](#)

Not documented.

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 Class](#) |
[TCustomVirtualStringTree.ComputeNodeHeight](#)
[Method](#) | [TCustomVirtualStringTree.Create](#)
[Constructor](#)

TCustomVirtualStringTree.ContentToClipboard Method

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
function ContentToClipboard(Format: Word; Source: TVirtualStringTree; C: TCustomVirtualStringTree): S;
function ContentToHTML(Source: TVSTTextSourceType; C: TCustomVirtualStringTree): S;
function ContentToRTF(Source: TVSTTextSourceType): S;
function ContentToText(Source: TVSTTextSourceType; S: string): S;
function ContentToUnicode(Source: TVSTTextSourceType): S;
```

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 Class](#) |
[TCustomVirtualStringTree.ContentToClipboard](#)
Method |
[TCustomVirtualStringTree.DefineProperties](#)
Method

TCustomVirtualStringTree.Create Constructor

[TCustomVirtualStringTree Class](#)

Constructor 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 Class](#) |
[TCustomVirtualStringTree.Create](#)
[Constructor](#) |
[TCustomVirtualStringTree.DoCreateEditor](#)
[Method](#)

TCustomVirtualStringTree.DefineProperties Method

[TCustomVirtualStringTree Class](#)

Helper method to customize loading and saving persistent tree data.

Pascal

```
procedure DefineProperties(Filer: TFile); 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 Class](#) |
[TCustomVirtualStringTree.DefineProperties](#)
Method |
[TCustomVirtualStringTree.DoGetNodeHint](#)
Method

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 Class](#) |
[TCustomVirtualStringTree.DoCreateEditor](#)
Method |
[TCustomVirtualStringTree.DoGetNodeTooltip](#)
Method

TCustomVirtualStringTree.DoGetNodeHint Method

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
function DoGetNodeHint(Node: PVirtualNode; Column: T
```

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 Class](#) |
[TCustomVirtualStringTree.DoGetNodeHint](#)
Method |
[TCustomVirtualStringTree.DoGetNodeWidth](#)
Method

TCustomVirtualStringTree.DoGetNodeTooltip Method

[TCustomVirtualStringTree 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

[TCustomVirtualStringTree Class](#)

Links

[TCustomVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.DoGetNodeTooltip](#)
Method |
[TCustomVirtualStringTree.DoGetText](#)
Method

TCustomVirtualStringTree.DoGetNodeWidth Method

[TCustomVirtualStringTree Class](#)

Overridable method which always returns 0.

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

[TCustomVirtualStringTree Class](#)

Links

[TCustomVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.DoGetNodeWidth](#)
Method |
[TCustomVirtualStringTree.DoIncrementalSearch](#)
Method

TCustomVirtualStringTree.DoGetText Method

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
procedure DoGetText(Node: PVirtualNode; Column: TCol
```

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 Class](#) |
[TCustomVirtualStringTree.DoGetText](#)
Method |
[TCustomVirtualStringTree.DoNewText](#)
Method

TCustomVirtualStringTree.DoIncrementalSearch Method

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
function DoIncrementalSearch(Node: PVirtualNode; con
```

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 Class](#) |
[TCustomVirtualStringTree.DoIncrementalSearch](#)
Method |
[TCustomVirtualStringTree.DoPaintNode](#) Method

TCustomVirtualStringTree.DoNewText Method

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
procedure DoNewText(Node: PVirtualNode; Column: TCol
```

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 Class](#) |
[TCustomVirtualStringTree.DoNewText](#)
Method |
[TCustomVirtualStringTree.DoPaintText](#)
Method

TCustomVirtualStringTree.DoPaintNode Method

[TCustomVirtualStringTree 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

[TCustomVirtualStringTree Class](#)

Links

[TCustomVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.DoPaintNode](#)
Method |
[TCustomVirtualStringTree.DoShortenString](#)
Method

TCustomVirtualStringTree.DoPaintText Method

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
procedure DoPaintText(Node: PVirtualNode; const Canv
```

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 Class](#) |
[TCustomVirtualStringTree.DoPaintText](#)
Method |
[TCustomVirtualStringTree.DoTextDrawing](#)
Method

TCustomVirtualStringTree.DoShortenStrir Method

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
function DoShortenString(Canvas: TCanvas; Node: PVir
```

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 Class](#) |
[TCustomVirtualStringTree.DoShortenString](#)
Method |
[TCustomVirtualStringTree.DoTextMeasuring](#)
Method

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 Class](#) |
[TCustomVirtualStringTree.DoTextDrawing](#)
Method |
[TCustomVirtualStringTree.GetOptionsClass](#)
Method

TCustomVirtualStringTree.DoTextMeasuri Method

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
function DoTextMeasuring(Canvas: TCanvas; Node: PVir
```

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 Class](#) |
[TCustomVirtualStringTree.DoTextMeasuring](#)
[Method](#) |
[TCustomVirtualStringTree.GetTextInfo](#)
[Method](#)

TCustomVirtualStringTree.GetOptionsClass Method

[TCustomVirtualStringTree Class](#)

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

Pascal

```
function GetOptionsClass: TTreeOptionsClass; override;
```

Description

GetOptionsClass is a special purpose method to return a certain class which is used by the tree for its options. TVirtualBaseTree always returns TCustomVirtualTreeOptions but descendants can override this method to return own classes.

For ease of use it makes much sense to always use the same name for the tree's options (which is TreeOptions). By using a customized options class, however, the wrong type is returned by this property. Hence it is meaningful to override TreeOptions and return the derived options class. To make this work the tree descendant must additionally provide new access methods for this property. An example can be seen in TVirtualStringTree:

```

TVirtualStringTree = class(TCustomVirtualStringTr
private
    function GetOptions: TStringTreeOptions;
    procedure SetOptions(const Value: TStringTreeOp
protected
    function GetOptionsClass: TTreeOptionsClass; ov
public
    property Canvas;
published
    ...
    property TreeOptions: TStringTreeOptions read G
    ...
end;

...

//----- TVirtualStringTree -----

function TVirtualStringTree.GetOptions: TStringTree

begin
    Result := FOptions as TStringTreeOptions;
end;

//-----

procedure TVirtualStringTree.SetOptions(const Value

begin
    FOptions.Assign(Value);
end;

//-----

function TVirtualStringTree.GetOptionsClass: TTreeO

```

```
begin
  Result := TStringTreeOptions;
end;
```

Class

[TCustomVirtualStringTree Class](#)

Links

[TCustomVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.GetOptionsClass](#)
Method |
[TCustomVirtualStringTree.InternalData](#)
Method

TCustomVirtualStringTree.GetTextInfo Method

[TCustomVirtualStringTree Class](#)

Helper method for node editors, hints etc.

Pascal

```
procedure GetTextInfo(Node: PVirtualNode; Column: TC
```

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 Class](#) |
[TCustomVirtualStringTree.GetTextInfo](#)
Method |
[TCustomVirtualStringTree.InvalidateNode](#)
Method

TCustomVirtualStringTree.InternalData Method

[TCustomVirtualStringTree Class](#) | [See Also](#)

Returns the address of the internal data for a tree class.

Pascal

```
function InternalData(Node: PVirtualNode): Pointer;
```

Description

In TBaseVirtualTreeView this method returns nil but should be overridden in descendants to allow proper access to the internal data of **Node** if the descendant tree has allocated internal data.

See Also

[Data handling](#)

Class

[TCustomVirtualStringTree Class](#)

Links

[TCustomVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.InternalData Method](#) |
[TCustomVirtualStringTree.MainColumnChanged Method](#)

TCustomVirtualStringTree.InvalidateNode Method

[TCustomVirtualStringTree Class](#)

Invalidates the given node.

Pascal

```
function InvalidateNode(Node: PVirtualNode): TRect;
```

Description

InvalidateNode initiates repaint of the given node by calling InvalidateRect with the node's display rectangel and returns this rectangle.

Class

[TCustomVirtualStringTree Class](#)

Links

[TCustomVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.InvalidateNode](#)
[Method](#) | [TCustomVirtualStringTree.Path](#)
[Method](#)

TCustomVirtualStringTree.MainColumnCh 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 Class](#) |
[TCustomVirtualStringTree.MainColumnChanged
Method](#) | [TCustomVirtualStringTree.ReadChunk
Method](#)

TCustomVirtualStringTree.Path Method

[TCustomVirtualStringTree Class](#)

Not documented.

Pascal

```
function Path(Node: PVirtualNode; Column: TColumnInd
```

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 Class](#) |
[TCustomVirtualStringTree.Path Method](#) |
[TCustomVirtualStringTree.ReadOldStringOptions Method](#)

TCustomVirtualStringTree.ReadChunk Method

[TCustomVirtualStringTree 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

[TCustomVirtualStringTree Class](#)

Links

[TCustomVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualStringTree Class](#) |
[TCustomVirtualStringTree.ReadChunk](#)
Method |
[TCustomVirtualStringTree.ReinitNode](#)
Method

TCustomVirtualStringTree.ReadOldString 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 Class](#) |
[TCustomVirtualStringTree.ReadOldStringOptions](#)
Method |
[TCustomVirtualStringTree.RenderOLEData](#)
Method

TCustomVirtualStringTree.ReinitNode Method

[TCustomVirtualStringTree Class](#)

Forces a reinitialization of the given node.

Pascal

```
procedure ReinitNode(Node: PVirtualNode; Recursive:
```

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 Class](#) |
[TCustomVirtualStringTree.ReinitNode](#)
[Method](#) |
[TCustomVirtualStringTree.WriteChunks](#)
[Method](#)

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

```
procedure WriteChunks(Stream: TStream; Node: PVirtua
```

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!](#)



[Classes](#) |
[TCustomVirtualStringTree Class](#) |
[TEnumFormatEtc Class](#)

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

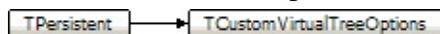
 public

 read only

 Method

 virtual

Class Hierarchy



File

VirtualTrees

Links

[Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TCustomVirtualTreeOptions Class |
TCustomVirtualTreeOptions.AutoOptions
Property

TCustomVirtualTreeOptions.AnimationOp Property

TCustomVirtualTreeOptions Class

Options related to animations.

Pascal

```
property AnimationOptions: TVTAnimationOptions;
```

Description

These options can be used to switch certain animation effects in a tree.

Class

TCustomVirtualTreeOptions Class

Links

TCustomVirtualTreeOptions Class

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualTreeOptions Class](#) |
[TCustomVirtualTreeOptions.AnimationOptions](#)
[Property](#) |
[TCustomVirtualTreeOptions.MiscOptions](#)
[Property](#)

TCustomVirtualTreeOptions.AutoOptions Property

[TCustomVirtualTreeOptions Class](#)

Options related to automatic actions.

Pascal

```
property AutoOptions: TVTAutoOptions;
```

Description

These options can be used to switch certain actions in a tree which happen automatically under certain circumstances.

Class

[TCustomVirtualTreeOptions Class](#)

Links

[TCustomVirtualTreeOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualTreeOptions Class](#) |
[TCustomVirtualTreeOptions.AutoOptions](#)
Property |
[TCustomVirtualTreeOptions.Owner](#)
Property

TCustomVirtualTreeOptions.MiscOptions Property

[TCustomVirtualTreeOptions Class](#)

Options not related to any other category.

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 Class](#) |
[TCustomVirtualTreeOptions.MiscOptions](#)
[Property](#) |
[TCustomVirtualTreeOptions.PaintOptions](#)
[Property](#)

TCustomVirtualTreeOptions.Owner Property

[TCustomVirtualTreeOptions Class](#)

Owner tree to which the property class belongs.

Pascal

```
property Owner: TBaseVirtualTree;
```

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 Class](#) |
[TCustomVirtualTreeOptions.Owner Property](#) |
[TCustomVirtualTreeOptions.SelectionOptions Property](#)

TCustomVirtualTreeOptions.PaintOptions Property

[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 Class](#) |
[TCustomVirtualTreeOptions.PaintOptions](#)
[Property](#) |
[TCustomVirtualTreeOptions.AssignTo](#)
[Method](#)

TCustomVirtualTreeOptions.SelectionOpt Property

[TCustomVirtualTreeOptions Class](#)

Options related to the way nodes can be selected.

Pascal

```
property SelectionOptions: TVTSelectionOptions;
```

Description

These options can be used to switch the way how nodes can be selected in a tree.

Class

[TCustomVirtualTreeOptions Class](#)

Links

[TCustomVirtualTreeOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TCustomVirtualTreeOptions Class](#) |
[TCustomVirtualTreeOptions.SelectionOptions
Property](#) | [TCustomVirtualTreeOptions.Create
Constructor](#)

TCustomVirtualTreeOptions.AssignTo Method

[TCustomVirtualTreeOptions Class](#)

Used to copy this option class to another option collection.

Pascal

```
procedure AssignTo(Dest: TPersistent); override;
```

Description

This is the usual method to support streaming or simply copying of this class. To stay open for future enhancements in form of new descendants not Assign but AssignTo has been used. AssignTo is called by TPersistent if there is no Assign method.

Class

[TCustomVirtualTreeOptions Class](#)

Links

[TCustomVirtualTreeOptions Class](#)

What do you think about this topic? [Send feedback!](#)



TCustomVirtualTreeOptions Class |
TCustomVirtualTreeOptions.AssignTo
Method

TCustomVirtualTreeOptions.Create Constructor

TCustomVirtualTreeOptions Class

Constructor of the class.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree); virtual
```

Description

Used to assign default values to all sub lists.

Class

TCustomVirtualTreeOptions Class

Links

TCustomVirtualTreeOptions Class

What do you think about this topic? [Send feedback!](#)



TEnumFormatEtc Class

[Classes](#) | [Methods](#) | [Legend](#)

```
TEnumFormatEtc = class(TInterfacedObject, IEnumForma
```

Group

[Classes](#)

Members

Methods

  [Clone](#)

Not documented.

  [Create](#)

Not documented.

  [Next](#)

Not documented.

  [Reset](#)

Not documented.

  [Skip](#)

Not documented.

Legend

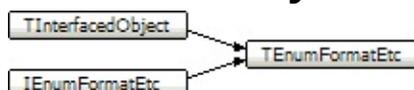


public



Method

Class Hierarchy



File

VirtualTrees

Links

[Classes](#), [Methods](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



[TEnumFormatEtc Class](#) |
[TEnumFormatEtc.Create](#)
[Constructor](#)

TEnumFormatEtc.Clone Method

[TEnumFormatEtc Class](#)

Not documented.

Pascal

```
function Clone(out Enum: IEnumFormatEtc): HRESULT; s
```

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 Class](#) |
[TEnumFormatEtc.Clone Method](#) |
[TEnumFormatEtc.Next Method](#)

TEnumFormatEtc.Create Constructor

[TEnumFormatEtc Class](#)

Not documented.

Pascal

```
constructor Create(Tree: TBaseVirtualTree; AFormatEt
```

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 Class](#) |
[TEnumFormatEtc.Create](#)
[Constructor](#) |
[TEnumFormatEtc.Reset Method](#)

TEnumFormatEtc.Next Method

[TEnumFormatEtc Class](#)

Not documented.

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 Class](#) |
[TEnumFormatEtc.Next Method](#) |
[TEnumFormatEtc.Skip Method](#)

TEnumFormatEtc.Reset Method

[TEnumFormatEtc Class](#)

Not documented.

Pascal

```
function Reset: HRESULT; stdcall;
```

Description

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

Class

[TEnumFormatEtc Class](#)

Links

[TEnumFormatEtc Class](#)

What do you think about this topic? [Send feedback!](#)



[TEnumFormatEtc Class](#) |
[TEnumFormatEtc.Reset Method](#)

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!](#)



[Classes](#) | [TEnumFormatEtc Class](#)
| [TStringEditLink Class](#)

TScrollBarOptions Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

```
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.

Legend



published



Property



public



Method



virtual



protected

Class Hierarchy



File

VirtualTrees

Links

[Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TScrollBarOptions.AlwaysVisible Property

[TScrollBarOptions Class](#)

Not documented.

Pascal

```
property AlwaysVisible: Boolean;
```

Description

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

Class

[TScrollBarOptions Class](#)

Links

[TScrollBarOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TScrollBarOptions Class](#) |
[TScrollBarOptions.AlwaysVisible](#)
Property |
[TScrollBarOptions.ScrollBars](#)
Property

TScrollBarOptions.HorizontalIncrement Property

[TScrollBarOptions Class](#)

Not documented.

Pascal

```
property HorizontalIncrement: TVTScrollIncrement;
```

Description

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

Class

[TScrollBarOptions Class](#)

Links

[TScrollBarOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TScrollBarOptions Class](#) |
[TScrollBarOptions.HorizontalIncrement](#)
[Property](#) |
[TScrollBarOptions.ScrollBarStyle](#)
[Property](#)

TScrollBarOptions.ScrollBars Property

[TScrollBarOptions Class](#)

Not documented.

Pascal

```
property ScrollBars: TScrollStyle;
```

Description

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

Class

[TScrollBarOptions Class](#)

Links

[TScrollBarOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TScrollBarOptions Class](#) |
[TScrollBarOptions.ScrollBars](#)
[Property](#) |
[TScrollBarOptions.VerticalIncrement](#)
[Property](#)

TScrollBarOptions.ScrollBarStyle Property

[TScrollBarOptions Class](#)

Not documented.

Pascal

```
property ScrollBarStyle: TScrollBarStyle;
```

Description

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

Class

[TScrollBarOptions Class](#)

Links

[TScrollBarOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TScrollBarOptions Class](#) |
[TScrollBarOptions.ScrollBarStyle](#)
[Property](#) |
[TScrollBarOptions.Assign Method](#)

TScrollBarOptions.VerticalIncrement Property

[TScrollBarOptions Class](#)

Not documented.

Pascal

```
property VerticalIncrement: TVTScrollIncrement;
```

Description

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

Class

[TScrollBarOptions Class](#)

Links

[TScrollBarOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TScrollBarOptions Class](#) |
[TScrollBarOptions.VerticalIncrement](#)
[Property](#) | [TScrollBarOptions.Create](#)
[Constructor](#)

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 Class](#) |
[TScrollBarOptions.Assign Method](#)
| [TScrollBarOptions.GetOwner Method](#)

TScrollBarOptions.Create Constructor

[TScrollBarOptions Class](#)

Not documented.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree);
```

Description

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

Class

[TScrollBarOptions Class](#)

Links

[TScrollBarOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TScrollBarOptions Class](#) |
[TScrollBarOptions.Create](#)
[Constructor](#)

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!](#)



[Classes](#) | [TScrollBarOptions Class](#)
| [TStringTreeOptions Class](#)

TStringEditLink Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

TStringEditLink is the standard node editor of a [TVirtualStringTree](#).

Pascal

```
TStringEditLink = class(TInterfacedObject, IVTEditLi
```

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

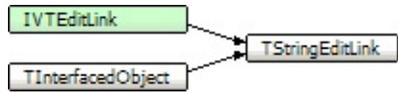


Method



virtual

Class Hierarchy



File

VirtualTrees

Links

[Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TStringEditLink.Edit Property

TStringEditLink Class

Not documented.

Pascal

```
property Edit: TVTEdit;
```

Description

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

Class

TStringEditLink Class

Links

TStringEditLink Class

What do you think about this topic? [Send feedback!](#)



[TStringEditLink Class](#) |
[TStringEditLink.Edit Property](#) |
[TStringEditLink.CancelEdit Method](#)

TStringEditLink.BeginEdit Method

[TStringEditLink Class](#)

This function will be called by the virtual string tree when the editing starts.

Pascal

```
function BeginEdit: Boolean; virtual; stdcall;
```

Description

Please see interface [IVTEditLink](#) for a detailed explanation of this interface function.

Class

[TStringEditLink Class](#)

Links

[TStringEditLink Class](#)

What do you think about this topic? [Send feedback!](#)



[TStringEditLink Class](#) |
[TStringEditLink.BeginEdit Method](#) |
[TStringEditLink.Create Constructor](#)

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

```
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 Class](#) |
[TStringEditLink.CancelEdit Method](#)
| [TStringEditLink.Destroy](#)
[Destructor](#)

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 Class](#) |
[TStringEditLink.Create Constructor](#)
| [TStringEditLink.EndEdit Method](#)

TStringEditLink.Destroy Destructor

[TStringEditLink Class](#)

Destructor of the class.

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 Class](#) |
[TStringEditLink.Destroy Destructor](#)
| [TStringEditLink.GetBounds](#)
[Method](#)

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 Class](#) |
[TStringEditLink.EndEdit Method](#) |
[TStringEditLink.PrepareEdit Method](#)

TStringEditLink.GetBounds Method

[TStringEditLink Class](#)

The virtual string tree uses this function to get the current bounding rect of the node editor.

Pascal

```
function GetBounds: TRect; virtual; stdcall;
```

Description

Please see interface [IVTEditLink](#) for a detailed explanation of this interface function.

Class

[TStringEditLink Class](#)

Links

[TStringEditLink Class](#)

What do you think about this topic? [Send feedback!](#)



[TStringEditLink Class](#) |
[TStringEditLink.GetBounds](#)
Method |
[TStringEditLink.ProcessMessage](#)
Method

TStringEditLink.PrepareEdit Method

[TStringEditLink Class](#)

This function is called by a virtual string tree to initialize the node editor.

Pascal

```
function PrepareEdit(Tree: TBaseVirtualTree; Node: P
```

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 Class](#) |
[TStringEditLink.PrepareEdit](#)
Method |
[TStringEditLink.SetBounds](#)
Method

TStringEditLink.ProcessMessage Method

[TStringEditLink Class](#)

This function is used to forward messages being directed to the virtual string tree.

Pascal

```
procedure ProcessMessage(var Message: TMessage); var
```

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!](#)



[Classes](#) | [TStringEditLink Class](#) | [TVirtualDrawTree Class](#)

TStringTreeOptions Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

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

```
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 Class](#) |
[TStringTreeOptions.AnimationOptions](#)
[Property](#) |
[TStringTreeOptions.MiscOptions](#)
[Property](#)

TStringTreeOptions.AutoOptions Property

[TStringTreeOptions Class](#)

Options related to automatic actions.

Pascal

```
property AutoOptions: TVTAutoOptions;
```

Description

These options can be used to switch certain actions in a tree which happen automatically under certain circumstances.

Class

[TStringTreeOptions Class](#)

Links

[TStringTreeOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TStringTreeOptions Class](#) |
[TStringTreeOptions.AutoOptions](#)
[Property](#) |
[TStringTreeOptions.PaintOptions](#)
[Property](#)

TStringTreeOptions.MiscOptions Property

[TStringTreeOptions Class](#)

Options not related to any other category.

Pascal

```
property MiscOptions: TVTMiscOptions;
```

Description

These options can be used to switch miscellaneous aspects in a tree.

Class

[TStringTreeOptions Class](#)

Links

[TStringTreeOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TStringTreeOptions Class](#) |
[TStringTreeOptions.MiscOptions](#)
[Property](#) |
[TStringTreeOptions.SelectionOptions](#)
[Property](#)

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 Class](#) |
[TStringTreeOptions.PaintOptions](#)
[Property](#) |
[TStringTreeOptions.StringOptions](#)
[Property](#)

TStringTreeOptions.SelectionOptions Property

[TStringTreeOptions Class](#)

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

[TStringTreeOptions Class](#)

The new options introduced by the class.

Pascal

```
property StringOptions: TVTStringOptions;
```

Description

StringOptions provides access to the newly introduced options by which the base class is extended.

Class

[TStringTreeOptions Class](#)

Links

[TStringTreeOptions Class](#)

What do you think about this topic? [Send feedback!](#)



TVirtualDrawTree Class

[Events](#) | [Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

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.
- 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 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.

●> **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.

●> **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.

●> **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](#)

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.
- ➤ **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.

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**
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 OnGetPopupMenu 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.

  **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 s 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.

 **HandleMouseDbIClick**

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

-  published
-  Property
-  public
-  protected
-  read only
-  Event
-  Method
-  virtual

Class Hierarchy



File

VirtualTrees

Links

[Events](#), [Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TVirtualDrawTree.Action Property

TVirtualDrawTree Class

Not documented.

Pascal

```
property Action;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Action Property](#) |
[TVirtualDrawTree.Alignment
Property](#)

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 Class](#) |
[TVirtualDrawTree.Align Property](#) |
[TVirtualDrawTree.Anchors](#)
[Property](#)

TVirtualDrawTree.Alignment Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Alignment](#)
[Property](#) |
[TVirtualDrawTree.AnimationDuration](#)
[Property](#)

TVirtualDrawTree.anchors Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property Anchors;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.anchors Property](#) |
[TVirtualDrawTree.AutoExpandDelay Property](#)

TVirtualDrawTree.AnimationDuration Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.AnimationDuration](#)
Property |
[TVirtualDrawTree.AutoScrollDelay](#)
Property

TVirtualDrawTree.AutoExpandDelay Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.AutoExpandDelay](#)
[Property](#) |
[TVirtualDrawTree.AutoScrollInterval](#)
[Property](#)

TVirtualDrawTree.AutoScrollDelay Property

[TVirtualDrawTree Class](#)

Time which determines when auto scrolling should start.

Pascal

```
property AutoScrollDelay: Cardinal;
```

Description

Once the mouse pointer has been moved near to a border a timer is started using the interval specified by AutoScrollDelay. When the timer has fired auto scrolling starts provided it is enabled (see also TreeOptions). The value is specified in milliseconds.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.AutoScrollDelay](#)
[Property](#) |
[TVirtualDrawTree.Background](#)
[Property](#)

TVirtualDrawTree.AutoScrollInterval Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.AutoScrollInterval](#)
[Property](#) |
[TVirtualDrawTree.BackgroundColorX](#)
[Property](#)

TVirtualDrawTree.BackgroundColor Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Background](#)
[Property](#) |
[TVirtualDrawTree.BackgroundColorOffsetY](#)
[Property](#)

TVirtualDrawTree.BackgroundColorOffsetX Property

[TVirtualDrawTree Class](#)

Horizontal offset of the background image.

Pascal

```
property BackgroundOffsetX: Integer;
```

Description

Determines the horizontal offset of the left border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.BackgroundColorX](#)
[Property](#) |
[TVirtualDrawTree.BevelEdges](#)
[Property](#)

TVirtualDrawTree.BackgroundColorY Property

[TVirtualDrawTree Class](#)

Vertical offset of the background image.

Pascal

```
property BackgroundOffsetY: Integer;
```

Description

Determines the vertical offset of the top border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.BackgroundColorY](#)
[Property](#) |
[TVirtualDrawTree.BevelInner](#)
[Property](#)

TVirtualDrawTree.BevelEdges Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property BevelEdges;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.BevelEdges](#)
[Property](#) |
[TVirtualDrawTree.BevelKind](#)
[Property](#)

TVirtualDrawTree.BevelInner Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.BevelInner
Property](#) |
[TVirtualDrawTree.BevelOuter
Property](#)

TVirtualDrawTree.BevelKind Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property BevelKind;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.BevelKind](#)
Property |
[TVirtualDrawTree.BevelWidth](#)
Property

TVirtualDrawTree.BevelOuter Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property BevelOuter;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.BevelOuter](#)
[Property](#) |
[TVirtualDrawTree.BiDiMode](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.BevelWidth](#)
Property |
[TVirtualDrawTree.BorderStyle](#)
Property

TVirtualDrawTree.BiDiMode Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property BiDiMode;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.BiDiMode](#)
[Property](#) |
[TVirtualDrawTree.BorderWidth](#)
[Property](#)

TVirtualDrawTree.BorderStyle Property

[TVirtualDrawTree Class](#)

Same as TForm.BorderStyle.

Pascal

```
property BorderStyle: TBorderStyle;
```

Description

See TForm.BorderStyle.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.BorderStyle](#)
[Property](#) |
[TVirtualDrawTree.ButtonFillMode](#)
[Property](#)

TVirtualDrawTree.BorderWidth Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property BorderWidth;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.BorderWidth](#)
[Property](#) |
[TVirtualDrawTree.ButtonStyle](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.ButtonFillMode](#)
[Property](#) |
[TVirtualDrawTree.Canvas Property](#)

TVirtualDrawTree.ButtonStyle Property

[TVirtualDrawTree Class](#)

Determines the look of node buttons.

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 Class](#) |
[TVirtualDrawTree.ButtonStyle](#)
[Property](#) |
[TVirtualDrawTree.ChangeDelay](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.Canvas Property](#) |
[TVirtualDrawTree.CheckImageKind Property](#)

TVirtualDrawTree.ChangeDelay Property

[TVirtualDrawTree Class](#)

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

Pascal

```
property ChangeDelay: Cardinal;
```

Description

In order to accumulate many quick changes in the tree you can use this delay value to specify after which wait time the OnChange event should occur. A value of 0 means to trigger OnChange immediately after the change (usually a selection or focus change) happend. Any value > 0 will start a timer which then triggers OnChange.

Note that there is the synchronous mode (started by BeginSynch) which effectively circumvents the change delay for the duration of the synchronous mode (stopped by EndSynch) regardless of the ChangeDelay setting.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.ChangeDelay](#)
[Property](#) |
[TVirtualDrawTree.ClipboardFormats](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.CheckImageKind
Property](#) | [TVirtualDrawTree.Color
Property](#)

TVirtualDrawTree.ClipboardFormats Property

[TVirtualDrawTree Class](#)

Special class to keep a list of clipboard format descriptions.

Pascal

```
property ClipboardFormats: TClipboardFormats;
```

Description

This TStringList descendant is used to keep a number of clipboard format descriptions, which are usually used to register clipboard formats with the system. Using a string list for this task allows to store enabled clipboard formats in the DFM.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.ClipboardFormats
Property](#) | [TVirtualDrawTree.Colors
Property](#)

TVirtualDrawTree.Color Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property Color;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Color Property](#) |
[TVirtualDrawTree.Constraints](#)
[Property](#)

TVirtualDrawTree.Colors Property

[TVirtualDrawTree Class](#)

A collection of colors used in the tree.

Pascal

```
property Colors: TVTColors;
```

Description

This property holds an instance of the [TVTColors](#) class, which is used to customize many of the colors used in a tree. Placing them all in a specialized class helps organizing the colors in the object inspector and improves general management.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Colors Property](#)
| [TVirtualDrawTree.Ctl3D Property](#)

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 Class](#) |
[TVirtualDrawTree.Constraints Property](#) |
[TVirtualDrawTree.CustomCheckImages Property](#)

TVirtualDrawTree.Ctl3D Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property Ctl3D;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Ctl3D Property](#) |
[TVirtualDrawTree.DefaultNodeHeight Property](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.CustomCheckImages](#)
[Property](#) |
[TVirtualDrawTree.DefaultPasteMode](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.DefaultNodeHeight](#)
[Property](#) |
[TVirtualDrawTree.DragCursor](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.DefaultPasteMode](#)
[Property](#) |
[TVirtualDrawTree.DragHeight](#)
[Property](#)

TVirtualDrawTree.DragCursor Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property DragCursor;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.DragCursor](#)
[Property](#) |
[TVirtualDrawTree.DragImageKind](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.DragHeight](#)
Property |
[TVirtualDrawTree.DragKind](#)
Property

TVirtualDrawTree.DragImageKind Property

[TVirtualDrawTree Class](#)

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

Pascal

```
property DragImageKind: TVTDragImageKind;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.DragImageKind](#)
[Property](#) |
[TVirtualDrawTree.DragMode](#)
[Property](#)

TVirtualDrawTree.DragKind Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.DragKind](#)
[Property](#) |
[TVirtualDrawTree.DragOperations](#)
[Property](#)

TVirtualDrawTree.DragMode Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property DragMode;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.DragMode](#)
[Property](#) |
[TVirtualDrawTree.DragType](#)
[Property](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.DragOperations](#)
[Property](#) |
[TVirtualDrawTree.DragWidth](#)
[Property](#)

TVirtualDrawTree.DragType Property

[TVirtualDrawTree Class](#)

Read or set which subsystem should be used for dragging.

Pascal

```
property DragType: TVTDragType;
```

Description

Traditionally, Delphi only supports its own drag mechanism, which is not compatible with the rest of the system. This VCL dragging also does not support to transport random data nor does it support drag operations between applications. Thus Virtual Treeview also supports the generally used OLE dragging, which in turn is incompatible with VCL dragging. Depending on your needs you can enable either VCL or OLE dragging as both together cannot be started. However, Virtual Treeview is able to act as drop target for both kind of data, independant of what is set in DragType.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.DragType Property](#) |
[TVirtualDrawTree.DrawSelectionMode Property](#)

TVirtualDrawTree.DragWidth Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.DragWidth](#)
Property |
[TVirtualDrawTree.EditDelay](#)
Property

TVirtualDrawTree.DrawSelectionMode Property

[TVirtualDrawTree Class](#)

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

Pascal

```
property DrawSelectionMode: TVTDrawSelectionMode;
```

Description

Virtual 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.DrawSelectionMode
Property](#) | [TVirtualDrawTree.Enabled
Property](#)

TVirtualDrawTree.EditDelay Property

[TVirtualDrawTree Class](#) | [See Also](#)

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

Pascal

```
property EditDelay: Cardinal;
```

Description

A node edit operation can be started using the keyboard (F2 key), in code using `EditNode` or by clicking twice on the same node (but not doing a double click). `EditDelay` is the 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 Class](#) |
[TVirtualDrawTree.EditDelay
Property](#) | [TVirtualDrawTree.Font
Property](#)

TVirtualDrawTree.Enabled Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property Enabled;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Enabled](#)
[Property](#) |
[TVirtualDrawTree.Header Property](#)

TVirtualDrawTree.Font Property

[TVirtualDrawTree Class](#)

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 Class](#) |
[TVirtualDrawTree.Font Property](#) |
[TVirtualDrawTree.HintAnimation Property](#)

TVirtualDrawTree.Header Property

[TVirtualDrawTree Class](#) | [See Also](#)

Provides access to the header instance.

Pascal

```
property Header : TVTHeader;
```

Description

This property is used to allow access to the header instance, which manages all aspects of the tree's header image as well as the column settings.

See Also

[TVTHeader](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Header Property](#)
| [TVirtualDrawTree.HintMode
Property](#)

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 Class](#) |
[TVirtualDrawTree.HintAnimation](#)
[Property](#) |
[TVirtualDrawTree.HotCursor](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.HintMode](#)
[Property](#) |
[TVirtualDrawTree.Images Property](#)

TVirtualDrawTree.HotCursor Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.HotCursor](#)
[Property](#) |
[TVirtualDrawTree.IncrementalSearch](#)
[Property](#)

TVirtualDrawTree.Images Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Images Property](#) |
[TVirtualDrawTree.IncrementalSearchDirection
Property](#)

TVirtualDrawTree.IncrementalSearch Property

[TVirtualDrawTree Class](#) | [See Also](#)

Read or set the current incremental search mode.

Pascal

```
property IncrementalSearch: TVTIncrementalSearch;
```

Description

Virtual Treeview can do an incremental search by calling back the application when comparing node captions. The `IncrementalSearch` property determines whether incremental search is enabled and which nodes should be searched through.

See Also

[IncrementalSearchDirection](#), [IncrementalSearchStart](#),
[IncrementalSearchTimeout](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.IncrementalSearch](#)
[Property](#) |
[TVirtualDrawTree.IncrementalSearchStart](#)
[Property](#)

TVirtualDrawTree.IncrementalSearchDirection Property

[TVirtualDrawTree Class](#) | [See Also](#)

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

Pascal

```
property IncrementalSearchDirection: TVTSearchDirection;
```

Description

When incremental search is enabled then Virtual Treeview can search forward and backward from the start point given by IncrementalSearchStart.

See Also

[IncrementalSearch](#), [IncrementalSearchStart](#),
[IncrementalSearchTime123out](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.IncrementalSearchDirection](#)
[Property](#) |
[TVirtualDrawTree.IncrementalSearchTimeout](#)
[Property](#)

TVirtualDrawTree.IncrementalSearchStart Property

[TVirtualDrawTree Class](#) | [See Also](#)

Read or set where to start incremental search.

Pascal

```
property IncrementalSearchStart: TVTSearchStart;
```

Description

When incremental search is enabled in the tree view then you can specify here, where to start the next incremental search operation from.

See Also

[IncrementalSearch](#), [IncrementalSearchDirection](#),
[IncrementalSearchTimeout](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.IncrementalSearchStart
Property](#) | [TVirtualDrawTree.Indent
Property](#)

TVirtualDrawTree.IncrementalSearchTime 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

```
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](#)

Links

[TVirtualDrawTree Class, See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.IncrementalSearchTimeout
Property](#) | [TVirtualDrawTree.LineMode
Property](#)

TVirtualDrawTree.Indent Property

[TVirtualDrawTree Class](#)

Read or set the indentation amount for node levels.

Pascal

```
property Indent: Cardinal;
```

Description

Each new level in the tree (child nodes of a parent node) are visually shifted to distinguish 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 Class](#) |
[TVirtualDrawTree.Indent Property](#) |
[TVirtualDrawTree.LineStyle Property](#)

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 Class](#) |
[TVirtualDrawTree.LineMode](#)
[Property](#) |
[TVirtualDrawTree.Margin Property](#)

TVirtualDrawTree.LineStyle Property

[TVirtualDrawTree Class](#)

Read or set the mode of the tree lines.

Pascal

```
property LineStyle: TVTLineStyle;
```

Description

Virtual Treeview allows to customize the lines used to display the node hierarchy. The default style is a dotted pattern, but you can also make solid lines or specify your own line pattern.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.LineStyle](#)
[Property](#) |
[TVirtualDrawTree.NodeAlignment](#)
[Property](#)

TVirtualDrawTree.Margin Property

[TVirtualDrawTree Class](#) | [See Also](#)

Read or set the tree's node margin.

Pascal

```
property Margin: Integer;
```

Description

The node margin is the distance between the cell bounds and its content like the lines, images, check box and so on. However this border is only applied to the left and right side of the node cell.

Note: there is also a `TextMargin` property in `TVirtualStringTree`, which is an additional border for the cell text only.

See Also

[TVirtualStringTree.TextMargin](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class, See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Margin Property](#)
| [TVirtualDrawTree.NodeDataSize
Property](#)

TVirtualDrawTree.NodeAlignment Property

[TVirtualDrawTree Class](#) | [See Also](#)

Read or set the node alignment value.

Pascal

```
property NodeAlignment: TVTNodeAlignment;
```

Description

Nodes have got an align member, which is used to determine the vertical position of the node's images and tree lines. The NodeAlignment property specifies how to interpret the value in the align member.

See Also

[TVirtualNode](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.NodeAlignment Property](#) |
[TVirtualDrawTree.OnAdvancedHeaderDraw Event](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.NodeDataSize](#)
[Property](#) |
[TVirtualDrawTree.OnAfterCellPaint](#)
[Event](#)

TVirtualDrawTree.OnAdvancedHeaderDraw Event

[TVirtualDrawTree Class](#) | [See Also](#)

Header paint support event.

Pascal

property OnAdvancedHeaderDraw: [TVTAdvancedHeaderPaint](#)

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 Class](#) |
[TVirtualDrawTree.OnAdvancedHeaderDraw
Event](#) | [TVirtualDrawTree.OnAfterItemErase
Event](#)

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 Class](#) |
[TVirtualDrawTree.OnAfterCellPaint](#)
[Event](#) |
[TVirtualDrawTree.OnAfterItemPaint](#)
[Event](#)

TVirtualDrawTree.OnAfterItemErase Event

[TVirtualDrawTree Class](#) | [See Also](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnAfterItemErase](#)
[Event](#) |
[TVirtualDrawTree.OnAfterPaint](#)
[Event](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnAfterItemPaint](#)
[Event](#) |
[TVirtualDrawTree.OnBeforeCellPaint](#)
[Event](#)

TVirtualDrawTree.OnAfterPaint Event

[TVirtualDrawTree Class](#) | [See Also](#)

Paint support event.

Pascal

```
property OnAfterPaint: TVTPaintEvent;
```

Description

Called after all nodes which needed an update have been drawn. This event is called once per paint cycle.

See Also

[Paint cycles and stages](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnAfterPaint Event](#) |
[TVirtualDrawTree.OnBeforeItemErase Event](#)

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 Class](#) |
[TVirtualDrawTree.OnBeforeCellPaint](#)
[Event](#) |
[TVirtualDrawTree.OnBeforeItemPaint](#)
[Event](#)

TVirtualDrawTree.OnBeforeItemErase Event

[TVirtualDrawTree Class](#) | [See Also](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnBeforeItemErase](#)
[Event](#) |
[TVirtualDrawTree.OnBeforePaint](#)
[Event](#)

TVirtualDrawTree.OnBeforeItemPaint Event

[TVirtualDrawTree Class](#) | [See Also](#)

Paint support event.

Pascal

```
property OnBeforeItemPaint: TVTBeforeItemPaintEvent;
```

Description

Called after the background of a node has been drawn and just before the node itself is painted. In this event the application gets the opportunity to decide whether a node should be drawn normally or should be skipped. The application can draw the node itself if necessary or leave the node area blank.

See Also

[Paint cycles and stages](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnBeforeItemPaint
Event](#) | [TVirtualDrawTree.OnChange
Event](#)

TVirtualDrawTree.OnBeforePaint Event

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnBeforePaint](#)
[Event](#) |
[TVirtualDrawTree.OnChecked](#)
[Event](#)

TVirtualDrawTree.OnChange Event

[TVirtualDrawTree Class](#)

Navigation support event.

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 Class](#) |
[TVirtualDrawTree.OnChange](#)
[Event](#) |
[TVirtualDrawTree.OnChecking](#)
[Event](#)

TVirtualDrawTree.OnChecked Event

[TVirtualDrawTree Class](#)

Check support event.

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 Class](#) |
[TVirtualDrawTree.OnChecked](#)
[Event](#) | [TVirtualDrawTree.OnClick](#)
[Property](#)

TVirtualDrawTree.OnChecking Event

[TVirtualDrawTree Class](#)

Check support event.

Pascal

```
property OnChecking: TVTCheckChangingEvent;
```

Description

Triggered when a node's check state is about to change and allows to prevent the change.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnChecking](#)
[Event](#) |
[TVirtualDrawTree.OnCollapsed](#)
[Event](#)

TVirtualDrawTree.OnClick Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnClick;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnClick](#)
[Property](#) |
[TVirtualDrawTree.OnCollapsing](#)
[Event](#)

TVirtualDrawTree.OnCollapsed Event

[TVirtualDrawTree Class](#)

Miscellaneous event.

Pascal

```
property OnCollapsed: TVTChangeEvent;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnCollapsed](#)
[Event](#) |
[TVirtualDrawTree.OnColumnClick](#)
[Event](#)

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 Class](#) |
[TVirtualDrawTree.OnCollapsing](#)
[Event](#) |
[TVirtualDrawTree.OnColumnDbClick](#)
[Event](#)

TVirtualDrawTree.OnColumnClick Event

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnColumnClick](#)
[Event](#) |
[TVirtualDrawTree.OnColumnResize](#)
[Event](#)

TVirtualDrawTree.OnColumnDbClick Event

[TVirtualDrawTree Class](#) | [See Also](#)

Header and column support event.

Pascal

```
property OnColumnDbClick: TVTColumnDbClickEvent;
```

Description

Same as OnColumnClick but for double clicks.

See Also

[OnColumnClick](#), [OnHeaderDbClick](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnColumnDbClick](#)
[Event](#) |
[TVirtualDrawTree.OnCompareNodes](#)
[Event](#)

TVirtualDrawTree.OnColumnResize Event

[TVirtualDrawTree Class](#)

Header and column support routine.

Pascal

```
property OnColumnResize: TVTHeaderNotifyEvent;
```

Description

Triggered when a column is being resized. During resize OnColumnResize is frequently hence you should make any code in the associated event handle a short and fast as possible.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnColumnResize](#)
[Event](#) |
[TVirtualDrawTree.OnCreateDataObject](#)
[Event](#)

TVirtualDrawTree.OnCompareNodes Event

[TVirtualDrawTree Class](#) | [See Also](#)

Sort and search support event.

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: TBaseV
  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 Class](#) |
[TVirtualDrawTree.OnCompareNodes](#)
[Event](#) |
[TVirtualDrawTree.OnCreateDragManager](#)
[Event](#)

TVirtualDrawTree.OnCreateDataObject Event

[TVirtualDrawTree Class](#)

Drag'n drop support event.

Pascal

```
property OnCreateDataObject: TVTCreateDataObjectEven
```

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 Class](#) |
[TVirtualDrawTree.OnCreateDataObject](#)
[Event](#) |
[TVirtualDrawTree.OnCreateEditor](#)
[Event](#)

TVirtualDrawTree.OnCreateDragManager Event

[TVirtualDrawTree Class](#)

Drag'n drop support event.

Pascal

```
property OnCreateDragManager: TVTCreateDragManagerEv
```

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 Class](#) |
[TVirtualDrawTree.OnCreateDragManager
Event](#) | [TVirtualDrawTree.OnDbClick
Property](#)

TVirtualDrawTree.OnCreateEditor Event

[TVirtualDrawTree Class](#) | [See Also](#)

Editing support event.

Pascal

```
property OnCreateEditor: TVTCreateEditorEvent;
```

Description

Allows to supply a customized node editor without changing the tree. TBaseVirtualTree triggers this event and raises an exception if there no editor is returned. If you don't want this then disable edit support for nodes in TreeOptions.MiscOptions. 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 Class](#) |
[TVirtualDrawTree.OnCreateEditor](#)
Event |
[TVirtualDrawTree.OnDragAllowed](#)
Event

TVirtualDrawTree.OnDbClick Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnDbClick;
```

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 Class](#) |
[TVirtualDrawTree.OnDbClick](#)
[Property](#) |
[TVirtualDrawTree.OnDragDrop](#)
[Event](#)

TVirtualDrawTree.OnDragAllowed Event

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnDragAllowed](#)
[Event](#) |
[TVirtualDrawTree.OnDragOver](#)
[Event](#)

TVirtualDrawTree.OnDragDrop Event

[TVirtualDrawTree Class](#)

Drag'n drop support event.

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.

```
procedure TMainForm.VTDragDrop(Sender: TBaseVirtual
  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 is recommend by Microsoft to order availa
// the first best format which we can accept is
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 TBaseV
          AttachMode := amInsertBefore
        else
          AttachMode := amNowhere;
        end;
        // in the case the drop target does an opti
        // to indicate this also to the drag source
        Sender.ProcessDrop(DataObject, Sender.DropT
        Break;
      end;
    end
  else
    begin
      // VCL drag'n drop, Effects contains by default
      // as usual the application has to find out wha
      Beep;
    end;
  end;
end;

```

Class

[TVirtualDrawTree Class](#)

Links

TVirtualDrawTree Class

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnDragDrop](#)
[Event](#) |
[TVirtualDrawTree.OnDrawHint](#)
[Event](#)

TVirtualDrawTree.OnDragOver Event

[TVirtualDrawTree Class](#) | [See Also](#)

Drag'n drop support event.

Pascal

```
property OnDragOver: TVTDragOverEvent;
```

Description

Triggered when Sender is the potential target of a drag'n drop operation. You can use this event to allow or deny a drop operation by setting Allowed to True or False, respectively. For conditions of OLE or VCL drag source see OnDragDrop.

See Also

[OnDragDrop](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnDragOver](#)
[Event](#) |
[TVirtualDrawTree.OnDrawNode](#)
[Event](#)

TVirtualDrawTree.OnDrawHint Event

[TVirtualDrawTree Class](#)

Triggered when a node hint or tooltip must be drawn.

Pascal

```
property OnDrawHint: TVTDrawHintEvent;
```

Description

Use an event handler for OnDrawHint to draw the hint or tooltip for the given node. You must implement this event and OnGetHintSize to get a hint at all.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnDrawHint](#)
[Event](#) |
[TVirtualDrawTree.OnEdited Event](#)

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 Class](#) |
[TVirtualDrawTree.OnDrawNode](#)
[Event](#) |
[TVirtualDrawTree.OnEditing Event](#)

TVirtualDrawTree.OnEdited Event

[TVirtualDrawTree Class](#) | [See Also](#)

Editing support event.

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnEdited Event](#)
| [TVirtualDrawTree.OnEndDock](#)
[Property](#)

TVirtualDrawTree.OnEditing Event

[TVirtualDrawTree Class](#) | [See Also](#)

Editing support event.

Pascal

```
property OnEditing: TVTEditChangingEvent;
```

Description

Triggered when a node is about to be edited. Use **Allowed** to allow or deny this action.

See Also

[Editors and editing](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnEditing Event](#)
| [TVirtualDrawTree.OnEndDrag](#)
Property

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 Class](#) |
[TVirtualDrawTree.OnEndDock](#)
[Property](#) |
[TVirtualDrawTree.OnEnter](#)
[Property](#)

TVirtualDrawTree.OnEndDrag Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnEndDrag;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnEndDrag](#)
[Property](#) |
[TVirtualDrawTree.OnExit Property](#)

TVirtualDrawTree.OnEnter Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnEnter;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnEnter](#)
[Property](#) |
[TVirtualDrawTree.OnExpanded](#)
[Event](#)

TVirtualDrawTree.OnExit Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnExit;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnExit Property](#)
| [TVirtualDrawTree.OnExpanding
Event](#)

TVirtualDrawTree.OnExpanded Event

[TVirtualDrawTree Class](#)

Misscellaneous event.

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 Class](#) |
[TVirtualDrawTree.OnExpanded](#)
[Event](#) |
[TVirtualDrawTree.OnFocusChanged](#)
[Event](#)

TVirtualDrawTree.OnExpanding Event

[TVirtualDrawTree Class](#)

Miscellaneous event.

Pascal

```
property OnExpanding: TVTChangingEvent;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnExpanding](#)
[Event](#) |
[TVirtualDrawTree.OnFocusChanging](#)
[Event](#)

TVirtualDrawTree.OnFocusChanged Event

[TVirtualDrawTree Class](#)

Navigation support event.

Pascal

```
property OnFocusChanged: TVTFocusChangeEvent;
```

Description

Triggered after the focused node changed. When examining **Node** keep in mind that it can be nil, meaning there is no focused node.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnFocusChanged](#)
[Event](#) |
[TVirtualDrawTree.OnFreeNode](#)
[Event](#)

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 Class](#) |
[TVirtualDrawTree.OnFocusChanging](#)
[Event](#) |
[TVirtualDrawTree.OnGetCellsEmpty](#)
[Event](#)

TVirtualDrawTree.OnFreeNode Event

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnFreeNode](#)
[Event](#) |
[TVirtualDrawTree.OnGetCursor](#)
[Event](#)

TVirtualDrawTree.OnGetCellsEmpty Event

[TVirtualDrawTree 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 OnGetCellsEmpty to let the application decide.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnGetCellsEmpty](#)
[Event](#) |
[TVirtualDrawTree.OnGetHeaderCursor](#)
[Event](#)

TVirtualDrawTree.OnGetCursor Event

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnGetCursor](#)
[Event](#) |
[TVirtualDrawTree.OnGetHelpContext](#)
[Event](#)

TVirtualDrawTree.OnGetHeaderCursor Event

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnGetHeaderCursor](#)
[Event](#) |
[TVirtualDrawTree.OnGetHintSize](#)
[Event](#)

TVirtualDrawTree.OnGetHelpContext Event

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnGetHelpContext](#)
[Event](#) |
[TVirtualDrawTree.OnGetImageIndex](#)
[Event](#)

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 Class](#) |
[TVirtualDrawTree.OnGetHintSize](#)
[Event](#) |
[TVirtualDrawTree.OnGetImageIndexEx](#)
[Event](#)

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 Class](#) |
[TVirtualDrawTree.OnGetImageIndex](#)
[Event](#) |
[TVirtualDrawTree.OnGetLineStyle](#)
[Event](#)

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 Class](#) |
[TVirtualDrawTree.OnGetImageIndexEx](#)
[Event](#) |
[TVirtualDrawTree.OnGetNodeDataSize](#)
[Event](#)

TVirtualDrawTree.OnGetLineStyle Event

[TVirtualDrawTree 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 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 Class](#) |
[TVirtualDrawTree.OnGetLineStyle](#)
[Event](#) |
[TVirtualDrawTree.OnGetNodeWidth](#)
[Event](#)

TVirtualDrawTree.OnGetNodeDataSize Event

[TVirtualDrawTree Class](#) | [See Also](#)

Data management event.

Pascal

```
property OnGetNodeDataSize: TVTGetNodeDataSizeEvent;
```

Description

Triggered when access to a node's data happens the first time but the actual data size is not yet set. Usually you would specify the size of the data you want to have added to each node by `NodeDataSize`, e.g. `SizeOf(TMyRecord)` is quite usual there (where `TMyRecord` is the structure you want to have stored in the node). Sometimes, however it is not possible to determine the node size in advance, so you can leave `NodeDataSize` being `-1` (the default value) and the `OnGetNodeDataSize` event is triggered as soon as the first regular node is created (the hidden root node does not have user data but internal data which is determined by other means).

See Also

`NodeDataSize`, [Data handling](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class, See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnGetNodeDataSize](#)
[Event](#) |
[TVirtualDrawTree.OnGetPopupMenu](#)
[Event](#)

TVirtualDrawTree.OnGetNodeWidth Event

[TVirtualDrawTree Class](#)

Triggered when a node is about to be drawn.

Pascal

```
property OnGetNodeWidth: TVTGetNodeWidthEvent;
```

Description

Use an event handler for OnGetNodeWidth to return your calculated width for the given node. Since the draw does not know the width of a node you have to tell it yourself.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnGetNodeWidth Event](#) |
[TVirtualDrawTree.OnGetUserClipboardFormats](#)
[Event](#)

TVirtualDrawTree.OnGetPopupMenu Event

[TVirtualDrawTree Class](#)

Miscellaneous event.

Pascal

```
property OnGetPopupMenu: TVTPopupEvent;
```

Description

This event allows the application to return a popup menu which is specific to a certain node. The tree does an automatic traversal all the way up to the top level node which is the parent of a given node to get a popup menu. If **Menu** is set then the traversal stops. Otherwise it continues until either a menu is set, AskParent is set to False or the top level parent has been reached.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnGetPopupMenu](#)
[Event](#) |
[TVirtualDrawTree.OnHeaderClick](#)
[Event](#)

TVirtualDrawTree.OnGetUserClipboardFo Event

[TVirtualDrawTree Class](#)

Drag'n drop and clipboard support event.

Pascal

```
property OnGetUserClipboardFormats: TVTGetUserClipbo
```

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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnGetUserClipboardFormats
Event](#) | [TVirtualDrawTree.OnHeaderDbClick
Event](#)

TVirtualDrawTree.OnHeaderClick Event

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnHeaderClick](#)
[Event](#) |
[TVirtualDrawTree.OnHeaderDragged](#)
[Event](#)

TVirtualDrawTree.OnHeaderDbClick Event

[TVirtualDrawTree Class](#) | [See Also](#)

Header & column support event.

Pascal

```
property OnHeaderDbClick: TVHeaderClickEvent;
```

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 Class](#) |
[TVirtualDrawTree.OnHeaderDbClick](#)
[Event](#) |
[TVirtualDrawTree.OnHeaderDraggedOut](#)
[Event](#)

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 Class](#) |
[TVirtualDrawTree.OnHeaderDragged](#)
[Event](#) |
[TVirtualDrawTree.OnHeaderDragging](#)
[Event](#)

TVirtualDrawTree.OnHeaderDraggedOut Event

[TVirtualDrawTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderDraggedOut: TVTHeaderDraggedOutEvent
```

Description

When during a header drag operation the mouse moves out of the header rectangle and the mouse button is released then an OnHeaderDraggedOut event will be fired with the target mouse position in screen coordinates.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnHeaderDraggedOut
Event](#) |
[TVirtualDrawTree.OnHeaderDraw Event](#)

TVirtualDrawTree.OnHeaderDragging Event

[TVirtualDrawTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderDragging: TVTHeaderDraggingEvent;
```

Description

Triggered just before dragging of a header button starts. Set **Allowed** to False if you want to prevent the drag operation of the given column.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnHeaderDragging Event](#) |
[TVirtualDrawTree.OnHeaderDrawQueryElements
Event](#)

TVirtualDrawTree.OnHeaderDraw Event

[TVirtualDrawTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderDraw: TVTHeaderPaintEvent;
```

Description

If you set the hoOwnerDraw style in TVTHeader.Options and a column has been set to vsOwnerDraw (see also TVirtualTreeColumn.Style) then OnDrawHeader is called whenever a column needs painting.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnHeaderDraw Event](#) |
[TVirtualDrawTree.OnHeaderMouseDown Event](#)

TVirtualDrawTree.OnHeaderDrawQueryElements Event

[TVirtualDrawTree Class](#) | [See Also](#)

Header & column support event.

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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnHeaderDrawQueryElements
Event](#) | [TVirtualDrawTree.OnHeaderMouseMove
Event](#)

TVirtualDrawTree.OnHeaderMouseDown Event

[TVirtualDrawTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderMouseDown: TVHeaderMouseEvent;
```

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 Class](#) |
[TVirtualDrawTree.OnHeaderMouseDown](#)
Event |
[TVirtualDrawTree.OnHeaderMouseUp](#)
Event

TVirtualDrawTree.OnHeaderMouseMove Event

[TVirtualDrawTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderMouseMove: TVTHHeaderMouseMoveEvent;
```

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 Class](#) |
[TVirtualDrawTree.OnHeaderMouseMove
Event](#) | [TVirtualDrawTree.OnHotChange
Event](#)

TVirtualDrawTree.OnHeaderMouseUp Event

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnHeaderMouseUp](#)
[Event](#) |
[TVirtualDrawTree.OnIncrementalSearch](#)
[Event](#)

TVirtualDrawTree.OnHotChange Event

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



TVirtualDrawTree Class |
TVirtualDrawTree.OnHotChange
Event |
TVirtualDrawTree.OnInitChildren
Event

TVirtualDrawTree.OnIncrementalSearch Event

TVirtualDrawTree Class

Miscellaneous event.

Pascal

```
property OnIncrementalSearch: TVTIncrementalSearchEv
```

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  
  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, No
end;

// By using StrLIComp we can specify a maximum le
// which match only partially.
Result := StrLIComp(PChar(S), PChar(PropText), Mi
end;

```

Notes

Usually incremental search allows to match also partially.
Hence it is recommended to do comparison only up to the
length

of the shorter string.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnIncrementalSearch
Event](#) | [TVirtualDrawTree.OnInitNode
Event](#)

TVirtualDrawTree.OnInitChildren Event

[TVirtualDrawTree Class](#) | [See Also](#)

Node management event.

Pascal

```
property OnInitChildren: TVTInitChildrenEvent;
```

Description

In order to allow the tree only to fill content where needed it is possible to set the `vsHasChildren` style in a node's 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 Class](#) |
[TVirtualDrawTree.OnInitChildren](#)
[Event](#) |
[TVirtualDrawTree.OnKeyAction](#)
[Event](#)

TVirtualDrawTree.OnInitNode Event

[TVirtualDrawTree Class](#) | [See Also](#)

Node management event.

Pascal

```
property OnInitNode: TVTInitNodeEvent;
```

Description

This event is important to connect the tree to your internal data. It is the ideal place to put references or whatever you need into a node's data area. You can set some initial states like selection, expansion state or that a node has child nodes.

See Also

[The virtual paradigm](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnInitNode](#)
[Event](#) |
[TVirtualDrawTree.OnKeyDown](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.OnKeyAction](#)
Event |
[TVirtualDrawTree.OnKeyPress](#)
Property

TVirtualDrawTree.OnKeyDown Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnKeyDown;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnKeyDown](#)
[Property](#) |
[TVirtualDrawTree.OnKeyUp](#)
[Property](#)

TVirtualDrawTree.OnKeyPress Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnKeyPress;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnKeyPress](#)
[Property](#) |
[TVirtualDrawTree.OnLoadNode](#)
[Event](#)

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 Class](#) |
[TVirtualDrawTree.OnKeyUp](#)
[Property](#) |
[TVirtualDrawTree.OnMeasureItem](#)
[Event](#)

TVirtualDrawTree.OnLoadNode Event

[TVirtualDrawTree Class](#) | [See Also](#)

Streaming support event.

Pascal

```
property OnLoadNode: TVTSaveNodeEvent;
```

Description

This event is typically triggered when serialized tree data must be restored, e.g. when loading the tree from file or stream or during a clipboard/drag'n drop operation. You should only read in what you wrote out in `OnSaveNode`. For safety there is a check in the loader code which tries to keep the internal serialization structure intact in case the application does not read correctly.

See Also

[OnSaveNode](#), [LoadFromStream](#), [SaveToStream](#),
[AddFromStream](#), [VTTTreeStreamVersion](#),
[TVTHeader.LoadFromStream](#), [TVTHeader.SaveToStream](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnLoadNode](#)
[Event](#) |
[TVirtualDrawTree.OnMouseDown](#)
[Property](#)

TVirtualDrawTree.OnMeasureItem Event

[TVirtualDrawTree Class](#) | [See Also](#)

Miscellaneous event.

Pascal

```
property OnMeasureItem: TVTMeasureItemEvent;
```

Description

Virtual Treeview supports individual node heights. However it might sometimes be unpractical to set this height in advance (e.g. during `OnInitNode`). Another scenario might be that multi line nodes must size themselves to accommodate 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 Class](#) |
[TVirtualDrawTree.OnMeasureItem](#)
[Event](#) |
[TVirtualDrawTree.OnMouseMove](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.OnMouseDown](#)
[Property](#) |
[TVirtualDrawTree.OnMouseUp](#)
[Property](#)

TVirtualDrawTree.OnMouseMove Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnMouseMove;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnMouseMove](#)
[Property](#) |
[TVirtualDrawTree.OnMouseWheel](#)
[Property](#)

TVirtualDrawTree.OnMouseUp Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnMouseUp;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnMouseUp](#)
[Property](#) |
[TVirtualDrawTree.OnNodeCopied](#)
[Event](#)

TVirtualDrawTree.OnMouseWheel Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnMouseWheel;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnMouseWheel](#)
[Property](#) |
[TVirtualDrawTree.OnNodeCopying](#)
[Event](#)

TVirtualDrawTree.OnNodeCopied Event

[TVirtualDrawTree Class](#)

Miscellaneous event.

Pascal

```
property OnNodeCopied: TVTNodeCopiedEvent;
```

Description

This event is triggered during drag'n drop after a node has been copied to a new location. Sender is the target tree where the copy operation took place.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnNodeCopied](#)
Event |
[TVirtualDrawTree.OnNodeMoved](#)
Event

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 Class](#) |
[TVirtualDrawTree.OnNodeCopying](#)
[Event](#) |
[TVirtualDrawTree.OnNodeMoving](#)
[Event](#)

TVirtualDrawTree.OnNodeMoved Event

[TVirtualDrawTree Class](#)

Miscellaneous event.

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!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnNodeMoved](#)
[Event](#) |
[TVirtualDrawTree.OnPaintBackground](#)
[Event](#)

TVirtualDrawTree.OnNodeMoving Event

[TVirtualDrawTree Class](#)

Miscellaneous event.

Pascal

```
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 Class](#) |
[TVirtualDrawTree.OnNodeMoving](#)
[Event](#) |
[TVirtualDrawTree.OnRenderOLEData](#)
[Event](#)

TVirtualDrawTree.OnPaintBackground Event

[TVirtualDrawTree Class](#)

Paint support event.

Pascal

```
property OnPaintBackground: TVTBackgroundPaintEvent;
```

Description

This event is triggered when the tree has finished its painting and there is an area which is not covered by nodes. For nodes there are various events to allow background customizat0n. 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 Class](#) |
[TVirtualDrawTree.OnPaintBackground](#)
[Event](#) |
[TVirtualDrawTree.OnResetNode](#)
[Event](#)

TVirtualDrawTree.OnRenderOLEData Event

[TVirtualDrawTree 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 descendants) have specified their own formats in `OnGetUserClipboardFormats`.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnRenderOLEData
Event](#) | [TVirtualDrawTree.OnResize
Property](#)

TVirtualDrawTree.OnResetNode Event

[TVirtualDrawTree Class](#) | [See Also](#)

Node management event.

Pascal

```
property OnResetNode: TVTChangeEvent;
```

Description

For large trees or simply because the content changed it is sometimes necessary to discard a certain node and release all its children. This can be done with `ResetNode` which will trigger this event.

See Also

`ResetNode`

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnResetNode](#)
Event |
[TVirtualDrawTree.OnSaveNode](#)
Event

TVirtualDrawTree.OnResize Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnResize;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnResize](#)
[Property](#) |
[TVirtualDrawTree.OnScroll Event](#)

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, VTTTreeStreamVersion,
TVTHeader.LoadFromStream, TVTHeader.SaveToStream

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnSaveNode](#)
[Event](#) |
[TVirtualDrawTree.OnShowScrollbar](#)
[Event](#)

TVirtualDrawTree.OnScroll Event

[TVirtualDrawTree Class](#) | [See Also](#)

Miscellaneous event.

Pascal

```
property OnScroll: TVTScrollEvent;
```

Description

This event is triggered when the tree is scrolled horizontally or vertically. You can use it to synchronize scrolling of several trees or other controls.

See Also

[OffsetXY](#)

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnScroll Event](#) |
[TVirtualDrawTree.OnStartDock](#)
Property

TVirtualDrawTree.OnShowScrollbar Event

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnShowScrollbar: TVTScrollbarShowEvent;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnShowScrollbar](#)
[Event](#) |
[TVirtualDrawTree.OnStateChange](#)
[Event](#)

TVirtualDrawTree.OnStartDock Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property OnStartDock;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnStartDock](#)
[Property](#) |
[TVirtualDrawTree.OnStructureChange](#)
[Event](#)

TVirtualDrawTree.OnStateChange Event

[TVirtualDrawTree Class](#)

Miscellaneous event.

Pascal

```
property OnStateChange: TVTStateChangeEvent;
```

Description

For special effects or in order to increase performance it is sometimes useful to know when the tree changes one of its internal states like `tsIncrementalSearching` or `tsOLEDDragging`. 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 Class](#) |
[TVirtualDrawTree.OnStateChange](#)
[Event](#) |
[TVirtualDrawTree.OnUpdating](#)
[Event](#)

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 Class](#) |
[TVirtualDrawTree.OnStructureChange](#)
[Event](#) |
[TVirtualDrawTree.ParentBiDiMode](#)
[Property](#)

TVirtualDrawTree.OnUpdating Event

[TVirtualDrawTree Class](#)

Miscellaneous event.

Pascal

```
property OnUpdating: TVTUpdatingEvent;
```

Description

This event is triggered when the application or the tree call `BeginUpdate` or `EndUpdate` and indicate so when a larger update operation takes place. This can for instance be used to show a hour glass wait cursor.

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.OnUpdating](#)
[Event](#) |
[TVirtualDrawTree.ParentColor](#)
[Property](#)

TVirtualDrawTree.ParentBiDiMode Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property ParentBiDiMode;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.ParentBiDiMode](#)
[Property](#) |
[TVirtualDrawTree.ParentCtl3D](#)
[Property](#)

TVirtualDrawTree.ParentColor Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property ParentColor;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.ParentColor](#)
[Property](#) |
[TVirtualDrawTree.ParentFont](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.ParentCtl3D](#)
[Property](#) |
[TVirtualDrawTree.ParentShowHint](#)
[Property](#)

TVirtualDrawTree.ParentFont 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 Class](#) |
[TVirtualDrawTree.ParentFont
Property](#) |
[TVirtualDrawTree.PopupMenu
Property](#)

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 Class](#) |
[TVirtualDrawTree.ParentShowHint](#)
Property |
[TVirtualDrawTree.RootNodeCount](#)
Property

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 Class](#) |
[TVirtualDrawTree.PopupMenu
Property](#) |
[TVirtualDrawTree.ScrollBarOptions
Property](#)

TVirtualDrawTree.RootNodeCount Property

[TVirtualDrawTree Class](#)

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 Class](#) |
[TVirtualDrawTree.RootNodeCount](#)
[Property](#) |
[TVirtualDrawTree.SelectionBlendFactor](#)
[Property](#)

TVirtualDrawTree.ScrollBarOptions Property

[TVirtualDrawTree 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.ScrollBarOptions](#)
[Property](#) |
[TVirtualDrawTree.SelectionCurveRadius](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.SelectionBlendFactor
Property](#) | [TVirtualDrawTree.ShowHint
Property](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.SelectionCurveRadius](#)
[Property](#) |
[TVirtualDrawTree.StateImages Property](#)

TVirtualDrawTree.ShowHint Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property ShowHint;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.ShowHint](#)
[Property](#) |
[TVirtualDrawTree.TabOrder](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.StateImages](#)
[Property](#) |
[TVirtualDrawTree.TabStop](#)
[Property](#)

TVirtualDrawTree.TabOrder Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property TabOrder;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.TabOrder](#)
[Property](#) |
[TVirtualDrawTree.TextMargin](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.TabStop](#)
[Property](#) |
[TVirtualDrawTree.TreeOptions](#)
[Property](#)

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 Class](#) |
[TVirtualDrawTree.TextMargin
Property](#) |
[TVirtualDrawTree.Visible Property](#)

TVirtualDrawTree.TreeOptions Property

[TVirtualDrawTree Class](#)

Reference to the tree's options.

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 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

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.TreeOptions](#)
[Property](#) |
[TVirtualDrawTree.WantTabs](#)
[Property](#)

TVirtualDrawTree.Visible Property

[TVirtualDrawTree Class](#)

Not documented.

Pascal

```
property Visible;
```

Description

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

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualDrawTree Class](#) |
[TVirtualDrawTree.Visible Property](#) |
[TVirtualDrawTree.GetOptionsClass](#)
Method

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 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

[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

```
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(TCustomVirtualStringTr
private
    function GetOptions: TStringTreeOptions;
    procedure SetOptions(const Value: TStringTreeOp
protected
    function GetOptionsClass: TTreeOptionsClass; ov
public
    property Canvas;
published
    ...
    property TreeOptions: TStringTreeOptions read G
    ...
end;

...

//----- TVirtualStringTree -----

function TVirtualStringTree.GetOptions: TStringTree

begin
    Result := FOptions as TStringTreeOptions;
end;

//-----

procedure TVirtualStringTree.SetOptions(const Value

begin
    FOptions.Assign(Value);
end;

//-----

function TVirtualStringTree.GetOptionsClass: TTreeO

begin

```

```
Result := TStringTreeOptions;  
end;
```

Class

[TVirtualDrawTree Class](#)

Links

[TVirtualDrawTree Class](#)

What do you think about this topic? [Send feedback!](#)



[Classes](#) | [TVirtualDrawTree Class](#) |
[TVirtualTreeColumn Class](#)

TVirtualStringTree Class

[Events](#) | [Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Descentant of [TBaseVirtualTree](#) which is able to manage node captions on its own.

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.
- **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.

🌐 **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.

●> **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.

●> **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.

●> **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.

➤ **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.

- ➤ **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.

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.

 **ColumnsIsEmpty**
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**
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.
- 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.

  **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 s 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.

   **HandleMouseDown**

Not documented.

   **HandleMouseUp**

Not documented.

   **HandleMouseDbClick**

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

-  published
-  Property



public



protected



read only



Event



Method



virtual

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

```
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 Class](#) |
[TVirtualStringTree.Action Property](#)
| [TVirtualStringTree.Alignment Property](#)

TVirtualStringTree.Align Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Align Property](#) |
[TVirtualStringTree.Anchors](#)
[Property](#)

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 Class](#) |
[TVirtualStringTree.Alignment](#)
[Property](#) |
[TVirtualStringTree.AnimationDuration](#)
[Property](#)

TVirtualStringTree.Anchors Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property Anchors;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.anchors Property](#) |
[TVirtualStringTree.AutoExpandDelay Property](#)

TVirtualStringTree.AnimationDuration Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.AnimationDuration](#)
[Property](#) |
[TVirtualStringTree.AutoScrollDelay](#)
[Property](#)

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 Class](#) |
[TVirtualStringTree.AutoExpandDelay](#)
[Property](#) |
[TVirtualStringTree.AutoScrollInterval](#)
[Property](#)

TVirtualStringTree.AutoScrollDelay Property

[TVirtualStringTree Class](#)

Time which determines when auto scrolling should start.

Pascal

```
property AutoScrollDelay: Cardinal;
```

Description

Once the mouse pointer has been moved near to a border a timer is started using the interval specified by AutoScrollDelay. When the timer has fired auto scrolling starts provided it is enabled (see also TreeOptions). The value is specified in milliseconds.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.AutoScrollDelay](#)
[Property](#) |
[TVirtualStringTree.Background](#)
[Property](#)

TVirtualStringTree.AutoScrollInterval Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.AutoScrollInterval](#)
[Property](#) |
[TVirtualStringTree.BackgroundColorX](#)
[Property](#)

TVirtualStringTree.BackgroundColor 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 Class](#) |
[TVirtualStringTree.Background](#)
[Property](#) |
[TVirtualStringTree.BackgroundColorY](#)
[Property](#)

TVirtualStringTree.BackgroundColorY Property

[TVirtualStringTree Class](#)

Horizontal offset of the background image.

Pascal

```
property BackgroundOffsetX: Integer;
```

Description

Determines the horizontal offset of the left border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.BackgroundColorX](#)
[Property](#) |
[TVirtualStringTree.BevelEdges](#)
[Property](#)

TVirtualStringTree.BackgroundColorY Property

[TVirtualStringTree Class](#)

Vertical offset of the background image.

Pascal

```
property BackgroundOffsetY: Integer;
```

Description

Determines the vertical offset of the top border of the background image. This value is relative to the target canvas where the tree is painted to (usually the tree window).

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.BackgroundColorY](#)
[Property](#) |
[TVirtualStringTree.BevelInner](#)
[Property](#)

TVirtualStringTree.BevelEdges Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property BevelEdges;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.BevelEdges](#)
[Property](#) |
[TVirtualStringTree.BevelKind](#)
[Property](#)

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 Class](#) |
[TVirtualStringTree.BevelInner
Property](#) |
[TVirtualStringTree.BevelOuter
Property](#)

TVirtualStringTree.BevelKind Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property BevelKind;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.BevelKind](#)
[Property](#) |
[TVirtualStringTree.BevelWidth](#)
[Property](#)

TVirtualStringTree.BevelOuter Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property BevelOuter;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.BevelOuter](#)
[Property](#) |
[TVirtualStringTree.BiDiMode](#)
[Property](#)

TVirtualStringTree.BevelWidth Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.BevelWidth](#)
[Property](#) |
[TVirtualStringTree.BorderStyle](#)
[Property](#)

TVirtualStringTree.BiDiMode Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property BiDiMode;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.BiDiMode](#)
[Property](#) |
[TVirtualStringTree.BorderWidth](#)
[Property](#)

TVirtualStringTree.BorderStyle Property

[TVirtualStringTree Class](#)

Same as TForm.BorderStyle.

Pascal

```
property BorderStyle: TBorderStyle;
```

Description

See TForm.BorderStyle.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.BorderStyle](#)
[Property](#) |
[TVirtualStringTree.ButtonFillMode](#)
[Property](#)

TVirtualStringTree.BorderStyle Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property BorderWidth;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.BorderWidth](#)
[Property](#) |
[TVirtualStringTree.ButtonStyle](#)
[Property](#)

TVirtualStringTree.ButtonFillMode Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.ButtonFillMode](#)
[Property](#) |
[TVirtualStringTree.Canvas](#)
[Property](#)

TVirtualStringTree.ButtonStyle Property

[TVirtualStringTree Class](#)

Determines the look of node buttons.

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 Class](#) |
[TVirtualStringTree.ButtonStyle](#)
[Property](#) |
[TVirtualStringTree.ChangeDelay](#)
[Property](#)

TVirtualStringTree.Canvas Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Canvas Property](#) |
[TVirtualStringTree.CheckImageKind Property](#)

TVirtualStringTree.ChangeDelay Property

[TVirtualStringTree Class](#)

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

Pascal

```
property ChangeDelay: Cardinal;
```

Description

In order to accumulate many quick changes in the tree you can use this delay value to specify after which wait time the OnChange event should occur. A value of 0 means to trigger OnChange immediately after the change (usually a selection or focus change) happend. Any value > 0 will start a timer which then triggers OnChange.

Note that there is the synchronous mode (started by BeginSynch) which effectively circumvents the change delay for the duration of the synchronous mode (stopped by EndSynch) regardless of the ChangeDelay setting.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.ChangeDelay](#)
[Property](#) |
[TVirtualStringTree.ClipboardFormats](#)
[Property](#)

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 Class](#) |
[TVirtualStringTree.CheckImageKind
Property](#) | [TVirtualStringTree.Color
Property](#)

TVirtualStringTree.ClipboardFormats Property

[TVirtualStringTree Class](#)

Special class to keep a list of clipboard format descriptions.

Pascal

```
property ClipboardFormats: TClipboardFormats;
```

Description

This TStringList descendant is used to keep a number of clipboard format descriptions, which are usually used to register clipboard formats with the system. Using a string list for this task allows to store enabled clipboard formats in the DFM.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.ClipboardFormats
Property](#) | [TVirtualStringTree.Colors
Property](#)

TVirtualStringTree.Color Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property Color;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Color Property](#) |
[TVirtualStringTree.Constraints](#)
[Property](#)

TVirtualStringTree.Colors Property

[TVirtualStringTree Class](#)

A collection of colors used in the tree.

Pascal

```
property Colors: TVTColors;
```

Description

This property holds an instance of the [TVTColors](#) class, which is used to customize many of the colors used in a tree. Placing them all in a specialized class helps organizing the colors in the object inspector and improves general management.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Colors Property](#)
| [TVirtualStringTree.Ctl3D Property](#)

TVirtualStringTree.Constraints Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Constraints Property](#) |
[TVirtualStringTree.CustomCheckImages Property](#)

TVirtualStringTree.Ctl3D Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property Ctl3D;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Ctl3D Property](#) |
[TVirtualStringTree.DefaultNodeHeight Property](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.CustomCheckImages](#)
[Property](#) |
[TVirtualStringTree.DefaultPasteMode](#)
[Property](#)

TVirtualStringTree.DefaultNodeHeight Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.DefaultNodeHeight](#)
[Property](#) |
[TVirtualStringTree.DefaultText](#)
[Property](#)

TVirtualStringTree.DefaultPasteMode Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.DefaultPasteMode](#)
[Property](#) |
[TVirtualStringTree.DragCursor](#)
[Property](#)

TVirtualStringTree.DefaultText Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property DefaultText: WideString;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.DefaultText](#)
[Property](#) |
[TVirtualStringTree.DragHeight](#)
[Property](#)

TVirtualStringTree.DragCursor Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property DragCursor;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.DragCursor](#)
[Property](#) |
[TVirtualStringTree.DragImageKind](#)
[Property](#)

TVirtualStringTree.DragHeight Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.DragHeight](#)
[Property](#) |
[TVirtualStringTree.DragKind](#)
[Property](#)

TVirtualStringTree.DragImageKind Property

[TVirtualStringTree Class](#)

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

Pascal

```
property DragImageKind: TVTDragImageKind;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.DragImageKind](#)
[Property](#) |
[TVirtualStringTree.DragMode](#)
[Property](#)

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 Class](#) |
[TVirtualStringTree.DragKind](#)
[Property](#) |
[TVirtualStringTree.DragOperations](#)
[Property](#)

TVirtualStringTree.DragMode Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property DragMode;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.DragMode](#)
[Property](#) |
[TVirtualStringTree.DragType](#)
[Property](#)

TVirtualStringTree.DragOperations Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.DragOperations](#)
[Property](#) |
[TVirtualStringTree.DragWidth](#)
[Property](#)

TVirtualStringTree.DragType Property

[TVirtualStringTree Class](#)

Read or set which subsystem should be used for dragging.

Pascal

```
property DragType: TVTDragType;
```

Description

Traditionally, Delphi only supports its own drag mechanism, which is not compatible with the rest of the system. This VCL dragging also does not support to transport random data nor does it support drag operations between applications. Thus Virtual Treeview also supports the generally used OLE dragging, which in turn is incompatible with VCL dragging. Depending on your needs you can enable either VCL or OLE dragging as both together cannot be started. However, Virtual Treeview is able to act as drop target for both kind of data, independant of what is set in DragType.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.DragType Property](#) |
[TVirtualStringTree.DrawSelectionMode Property](#)

TVirtualStringTree.DragWidth Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.DragWidth](#)
Property |
[TVirtualStringTree.EditDelay](#)
Property

TVirtualStringTree.DrawSelectionMode Property

[TVirtualStringTree Class](#)

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

Pascal

```
property DrawSelectionMode: TVTDrawSelectionMode;
```

Description

Virtual 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 Class](#) |
[TVirtualStringTree.DrawSelectionMode
Property](#) | [TVirtualStringTree.Enabled
Property](#)

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

```
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 Class](#) |
[TVirtualStringTree.EditDelay
Property](#) | [TVirtualStringTree.Font
Property](#)

TVirtualStringTree.Enabled Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property Enabled;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Enabled](#)
Property |
[TVirtualStringTree.Header](#)
Property

TVirtualStringTree.Font Property

[TVirtualStringTree Class](#)

Same as TWinControl.Font.

Pascal

```
property Font;
```

Description

See TWinControl.Font.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Font Property](#) |
[TVirtualStringTree.HintAnimation
Property](#)

TVirtualStringTree.Header Property

[TVirtualStringTree Class](#) | [See Also](#)

Provides access to the header instance.

Pascal

```
property Header : TVTHeader;
```

Description

This property is used to allow access to the header instance, which manages all aspects of the tree's header image as well as the column settings.

See Also

[TVTHeader](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Header](#)
[Property](#) |
[TVirtualStringTree.HintMode](#)
[Property](#)

TVirtualStringTree.HintAnimation Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.HintAnimation](#)
[Property](#) |
[TVirtualStringTree.HotCursor](#)
[Property](#)

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 Class](#) |
[TVirtualStringTree.HintMode](#)
[Property](#) |
[TVirtualStringTree.Images](#)
[Property](#)

TVirtualStringTree.HotCursor Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.HotCursor](#)
[Property](#) |
[TVirtualStringTree.IncrementalSearch](#)
[Property](#)

TVirtualStringTree.Images Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Images Property](#) |
[TVirtualStringTree.IncrementalSearchDirection
Property](#)

TVirtualStringTree.IncrementalSearch Property

[TVirtualStringTree Class](#) | [See Also](#)

Read or set the current incremental search mode.

Pascal

```
property IncrementalSearch: TVTIncrementalSearch;
```

Description

Virtual Treeview can do an incremental search by calling back the application when comparing node captions. The `IncrementalSearch` property determines whether incremental search is enabled and which nodes should be searched through.

See Also

[IncrementalSearchDirection](#), [IncrementalSearchStart](#),
[IncrementalSearchTimeout](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.IncrementalSearch](#)
[Property](#) |
[TVirtualStringTree.IncrementalSearchStart](#)
[Property](#)

TVirtualStringTree.IncrementalSearchDire Property

[TVirtualStringTree Class](#) | [See Also](#)

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

Pascal

```
property IncrementalSearchDirection: TVTSearchDirect
```

Description

When incremental search is enabled then Virtual Treeview can search forward and backward from the start point given by IncrementalSearchStart.

See Also

[IncrementalSearch](#), [IncrementalSearchStart](#),
[IncrementalSearchTime123out](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.IncrementalSearchDirection](#)
[Property](#) |
[TVirtualStringTree.IncrementalSearchTimeout](#)
[Property](#)

TVirtualStringTree.IncrementalSearchStart Property

[TVirtualStringTree Class](#) | [See Also](#)

Read or set where to start incremental search.

Pascal

```
property IncrementalSearchStart: TVTSearchStart;
```

Description

When incremental search is enabled in the tree view then you can specify here, where to start the next incremental search operation from.

See Also

[IncrementalSearch](#), [IncrementalSearchDirection](#),
[IncrementalSearchTimeout](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.IncrementalSearchStart
Property](#) | [TVirtualStringTree.Indent
Property](#)

TVirtualStringTree.IncrementalSearchTim 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 Class](#) |
[TVirtualStringTree.IncrementalSearchTimeout
Property](#) | [TVirtualStringTree.LineMode
Property](#)

TVirtualStringTree.Indent Property

[TVirtualStringTree Class](#)

Read or set the indentation amount for node levels.

Pascal

```
property Indent: Cardinal;
```

Description

Each new level in the tree (child nodes of a parent node) are visually shifted to distinguish 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 Class](#) |
[TVirtualStringTree.Indent Property](#)
| [TVirtualStringTree.LineStyle Property](#)

TVirtualStringTree.LineMode Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.LineMode](#)
[Property](#) |
[TVirtualStringTree.Margin](#) Property

TVirtualStringTree.LineStyle Property

[TVirtualStringTree Class](#)

Read or set the mode of the tree lines.

Pascal

```
property LineStyle: TVTLineStyle;
```

Description

Virtual Treeview allows to customize the lines used to display the node hierarchy. The default style is a dotted pattern, but you can also make solid lines or specify your own line pattern.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.LineStyle](#)
[Property](#) |
[TVirtualStringTree.NodeAlignment](#)
[Property](#)

TVirtualStringTree.Margin Property

[TVirtualStringTree Class](#) | [See Also](#)

Read or set the tree's node margin.

Pascal

```
property Margin: Integer;
```

Description

The node margin is the distance between the cell bounds and its content like the lines, images, check box and so on. However this border is only applied to the left and right side of the node cell.

Note: there is also a `TextMargin` property in `TVirtualStringTree`, which is an additional border for the cell text only.

See Also

[TVirtualStringTree.TextMargin](#)

Class

[TVirtualStringTree Class](#)

Links

TVirtualStringTree Class, See Also

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Margin Property](#)
| [TVirtualStringTree.NodeDataSize
Property](#)

TVirtualStringTree.NodeAlignment Property

[TVirtualStringTree Class](#) | [See Also](#)

Read or set the node alignment value.

Pascal

```
property NodeAlignment: TVTNodeAlignment;
```

Description

Nodes have got an align member, which is used to determine the vertical position of the node's images and tree lines. The NodeAlignment property specifies how to interpret the value in the align member.

See Also

[TVirtualNode](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.NodeAlignment Property](#) |
[TVirtualStringTree.OnAdvancedHeaderDraw Event](#)

TVirtualStringTree.NodeDataSize Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.NodeDataSize](#)
[Property](#) |
[TVirtualStringTree.OnAfterCellPaint](#)
[Event](#)

TVirtualStringTree.OnAdvancedHeaderDraw Event

[TVirtualStringTree Class](#) | [See Also](#)

Header paint support event.

Pascal

```
property OnAdvancedHeaderDraw: TVTAdvancedHeaderPaint
```

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 Class](#) |
[TVirtualStringTree.OnAdvancedHeaderDraw
Event](#) | [TVirtualStringTree.OnAfterItemErase
Event](#)

TVirtualStringTree.OnAfterCellPaint Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnAfterCellPaint
Event](#) |
[TVirtualStringTree.OnAfterItemPaint
Event](#)

TVirtualStringTree.OnAfterItemErase Event

[TVirtualStringTree Class](#) | [See Also](#)

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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnAfterItemErase](#)
[Event](#) |
[TVirtualStringTree.OnAfterPaint](#)
[Event](#)

TVirtualStringTree.OnAfterItemPaint Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnAfterItemPaint](#)
[Event](#) |
[TVirtualStringTree.OnBeforeCellPaint](#)
[Event](#)

TVirtualStringTree.OnAfterPaint Event

[TVirtualStringTree Class](#) | [See Also](#)

Paint support event.

Pascal

```
property OnAfterPaint: TVTPaintEvent;
```

Description

Called after all nodes which needed an update have been drawn. This event is called once per paint cycle.

See Also

[Paint cycles and stages](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnAfterPaint Event](#) |
[TVirtualStringTree.OnBeforeItemErase Event](#)

TVirtualStringTree.OnBeforeCellPaint Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnBeforeCellPaint](#)
[Event](#) |
[TVirtualStringTree.OnBeforeItemPaint](#)
[Event](#)

TVirtualStringTree.OnBeforeItemErase Event

[TVirtualStringTree Class](#) | [See Also](#)

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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnBeforeItemErase](#)
[Event](#) |
[TVirtualStringTree.OnBeforePaint](#)
[Event](#)

TVirtualStringTree.OnBeforeItemPaint Event

[TVirtualStringTree Class](#) | [See Also](#)

Paint support event.

Pascal

```
property OnBeforeItemPaint: TVTBeforeItemPaintEvent;
```

Description

Called after the background of a node has been drawn and just before the node itself is painted. In this event the application gets the opportunity to decide whether a node should be drawn normally or should be skipped. The application can draw the node itself if necessary or leave the node area blank.

See Also

[Paint cycles and stages](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnBeforeItemPaint
Event](#) | [TVirtualStringTree.OnChange
Event](#)

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 Class](#) |
[TVirtualStringTree.OnBeforePaint](#)
[Event](#) |
[TVirtualStringTree.OnChecked](#)
[Event](#)

TVirtualStringTree.OnChange Event

[TVirtualStringTree Class](#)

Navigation support event.

Pascal

```
property OnChange: 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 Class](#) |
[TVirtualStringTree.OnChange](#)
[Event](#) |
[TVirtualStringTree.OnChecking](#)
[Event](#)

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 Class](#) |
[TVirtualStringTree.OnChecked](#)
[Event](#) | [TVirtualStringTree.OnClick](#)
[Property](#)

TVirtualStringTree.OnChecking Event

[TVirtualStringTree Class](#)

Check support event.

Pascal

```
property OnChecking: TVTCheckChangingEvent;
```

Description

Triggered when a node's check state is about to change and allows to prevent the change.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnChecking](#)
[Event](#) |
[TVirtualStringTree.OnCollapsed](#)
[Event](#)

TVirtualStringTree.OnClick Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnClick;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnClick](#)
[Property](#) |
[TVirtualStringTree.OnCollapsing](#)
[Event](#)

TVirtualStringTree.OnCollapsed Event

[TVirtualStringTree Class](#)

Miscellaneous event.

Pascal

```
property OnCollapsed: TVTChangeEvent;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnCollapsed](#)
[Event](#) |
[TVirtualStringTree.OnColumnClick](#)
[Event](#)

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 Class](#) |
[TVirtualStringTree.OnCollapsing
Event](#) |
[TVirtualStringTree.OnColumnDbClick
Event](#)

TVirtualStringTree.OnColumnClick Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnColumnClick](#)
[Event](#) |
[TVirtualStringTree.OnColumnResize](#)
[Event](#)

TVirtualStringTree.OnColumnDbClick Event

[TVirtualStringTree Class](#) | [See Also](#)

Header and column support event.

Pascal

```
property OnColumnDbClick: TVTColumnDbClickEvent;
```

Description

Same as OnColumnClick but for double clicks.

See Also

[OnColumnClick](#), [OnHeaderDbClick](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnColumnDbClick](#)
[Event](#) |
[TVirtualStringTree.OnCompareNodes](#)
[Event](#)

TVirtualStringTree.OnColumnResize Event

[TVirtualStringTree Class](#)

Header and column support routine.

Pascal

```
property OnColumnResize: TVTHeaderNotifyEvent;
```

Description

Triggered when a column is being resized. During resize OnColumnResize is frequently hence you should make any code in the associated event handle a short and fast as possible.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnColumnResize](#)
[Event](#) |
[TVirtualStringTree.OnCreateDataObject](#)
[Event](#)

TVirtualStringTree.OnCompareNodes Event

[TVirtualStringTree Class](#) | [See Also](#)

Sort and search support event.

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: TBaseV
  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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnCompareNodes](#)
[Event](#) |
[TVirtualStringTree.OnCreateDragManager](#)
[Event](#)

TVirtualStringTree.OnCreateDataObject Event

[TVirtualStringTree Class](#)

Drag'n drop support event.

Pascal

```
property OnCreateDataObject: TVTCreateDataObjectEvent
```

Description

This event is called when the tree's drag manager needs a data object interface to start a drag'n drop operation. 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 Class](#) |
[TVirtualStringTree.OnCreateDataObject](#)
[Event](#) |
[TVirtualStringTree.OnCreateEditor](#)
[Event](#)

TVirtualStringTree.OnCreateDragManager Event

[TVirtualStringTree Class](#)

Drag'n drop support event.

Pascal

```
property OnCreateDragManager: TVTCreateDragManagerEv
```

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 Class](#) |
[TVirtualStringTree.OnCreateDragManager
Event](#) | [TVirtualStringTree.OnDbClick
Property](#)

TVirtualStringTree.OnCreateEditor Event

[TVirtualStringTree Class](#) | [See Also](#)

Editing support event.

Pascal

```
property OnCreateEditor: TVTCreateEditorEvent;
```

Description

Allows to supply a customized node editor without changing the tree. TBaseVirtualTree triggers this event and raises an exception if there no editor is returned. If you don't want this then disable edit support for nodes in TreeOptions.MiscOptions. 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 Class](#) |
[TVirtualStringTree.OnCreateEditor](#)
[Event](#) |
[TVirtualStringTree.OnDragAllowed](#)
[Event](#)

TVirtualStringTree.OnDbClick Property

[TVirtualStringTree Class](#)

Not documented.

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 Class](#) |
[TVirtualStringTree.OnDbClick](#)
Property |
[TVirtualStringTree.OnDragDrop](#)
Event

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 Class](#) |
[TVirtualStringTree.OnDragAllowed](#)
[Event](#) |
[TVirtualStringTree.OnDragOver](#)
[Event](#)

TVirtualStringTree.OnDragDrop Event

[TVirtualStringTree Class](#)

Drag'n drop support event.

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.

```
procedure TMainForm.VTDragDrop(Sender: TBaseVirtual  
    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 is recommend by Microsoft to order availa
// the first best format which we can accept is
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 TBaseV
          AttachMode := amInsertBefore
        else
          AttachMode := amNowhere;
        end;
        // in the case the drop target does an opti
        // to indicate this also to the drag source
        Sender.ProcessDrop(DataObject, Sender.DropT
        Break;
      end;
    end
  else
    begin
      // VCL drag'n drop, Effects contains by default
      // as usual the application has to find out wha
      Beep;
    end;
  end;
end;

```

Class

TVirtualStringTree Class

Links

TVirtualStringTree Class

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnDragDrop](#)
[Event](#) |
[TVirtualStringTree.OnEditCancelled](#)
[Event](#)

TVirtualStringTree.OnDragOver Event

[TVirtualStringTree Class](#) | [See Also](#)

Drag'n drop support event.

Pascal

```
property OnDragOver: TVTDragOverEvent;
```

Description

Triggered when Sender is the potential target of a drag'n drop operation. You can use this event to allow or deny a drop operation by setting Allowed to True or False, respectively. For conditions of OLE or VCL drag source see OnDragDrop.

See Also

[OnDragDrop](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnDragOver
Event](#) |
[TVirtualStringTree.OnEdited Event](#)

TVirtualStringTree.OnEditCancelled Event

[TVirtualStringTree Class](#) | [See Also](#)

Editing support event.

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 Class](#) |
[TVirtualStringTree.OnEditCancelled](#)
[Event](#) |
[TVirtualStringTree.OnEditing Event](#)

TVirtualStringTree.OnEdited Event

[TVirtualStringTree Class](#) | [See Also](#)

Editing support event.

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 Class](#) |
[TVirtualStringTree.OnEdited Event](#)
| [TVirtualStringTree.OnEndDock](#)
[Property](#)

TVirtualStringTree.OnEditing Event

[TVirtualStringTree Class](#) | [See Also](#)

Editing support event.

Pascal

```
property OnEditing: TVTEditChangingEvent;
```

Description

Triggered when a node is about to be edited. Use **Allowed** to allow or deny this action.

See Also

[Editors and editing](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnEditing](#)
[Event](#) |
[TVirtualStringTree.OnEndDrag](#)
[Property](#)

TVirtualStringTree.OnEndDock Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnEndDock](#)
[Property](#) |
[TVirtualStringTree.OnEnter](#)
[Property](#)

TVirtualStringTree.OnEndDrag Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnEndDrag;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnEndDrag](#)
[Property](#) |
[TVirtualStringTree.OnExit Property](#)

TVirtualStringTree.OnEnter Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnEnter;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnEnter](#)
[Property](#) |
[TVirtualStringTree.OnExpanded](#)
[Event](#)

TVirtualStringTree.OnExit Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnExit;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnExit Property](#)
| [TVirtualStringTree.OnExpanding
Event](#)

TVirtualStringTree.OnExpanded Event

[TVirtualStringTree Class](#)

Misscellaneous event.

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 Class](#) |
[TVirtualStringTree.OnExpanded](#)
[Event](#) |
[TVirtualStringTree.OnFocusChanged](#)
[Event](#)

TVirtualStringTree.OnExpanding Event

[TVirtualStringTree Class](#)

Miscellaneous event.

Pascal

```
property OnExpanding: TVTChangingEvent;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnExpanding](#)
[Event](#) |
[TVirtualStringTree.OnFocusChanging](#)
[Event](#)

TVirtualStringTree.OnFocusChanged Event

[TVirtualStringTree Class](#)

Navigation support event.

Pascal

```
property OnFocusChanged: TVTFocusChangeEvent;
```

Description

Triggered after the focused node changed. When examining **Node** keep in mind that it can be nil, meaning there is no focused node.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnFocusChanged](#)
[Event](#) |
[TVirtualStringTree.OnFreeNode](#)
[Event](#)

TVirtualStringTree.OnFocusChanging Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnFocusChanging](#)
[Event](#) |
[TVirtualStringTree.OnGetCellsEmpty](#)
[Event](#)

TVirtualStringTree.OnFreeNode Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnFreeNode](#)
[Event](#) |
[TVirtualStringTree.OnGetCursor](#)
[Event](#)

TVirtualStringTree.OnGetCellsEmpty 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 OnGetCellsEmpty to let the application decide.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnGetCellsEmpty](#)
[Event](#) |
[TVirtualStringTree.OnGetHeaderCursor](#)
[Event](#)

TVirtualStringTree.OnGetCursor Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnGetCursor](#)
[Event](#) |
[TVirtualStringTree.OnGetHelpContext](#)
[Event](#)

TVirtualStringTree.OnGetHeaderCursor Event

[TVirtualStringTree Class](#)

Header and column support event.

Pascal

```
property OnGetHeaderCursor: TVTGetHeaderCursorEvent;
```

Description

This event is triggered from the WM_SETCURSORS 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 Class](#) |
[TVirtualStringTree.OnGetHeaderCursor
Event](#) | [TVirtualStringTree.OnGetHint
Event](#)

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 Class](#) |
[TVirtualStringTree.OnGetHelpContext](#)
[Event](#) |
[TVirtualStringTree.OnGetImageIndex](#)
[Event](#)

TVirtualStringTree.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 Class](#) |
[TVirtualStringTree.OnGetHint Event](#) |
[TVirtualStringTree.OnGetImageIndexEx
Event](#)

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 Class](#) |
[TVirtualStringTree.OnGetImageIndex](#)
[Event](#) |
[TVirtualStringTree.OnGetLineStyle](#)
[Event](#)

TVirtualStringTree.OnGetImageIndexEx Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnGetImageIndexEx](#)
[Event](#) |
[TVirtualStringTree.OnGetNodeDataSize](#)
[Event](#)

TVirtualStringTree.OnGetLineStyle Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnGetLineStyle](#)
[Event](#) |
[TVirtualStringTree.OnGetPopupMenu](#)
[Event](#)

TVirtualStringTree.OnGetNodeDataSize Event

[TVirtualStringTree Class](#) | [See Also](#)

Data management event.

Pascal

```
property OnGetNodeDataSize: TVTGetNodeDataSizeEvent;
```

Description

Triggered when access to a node's data happens the first time but the actual data size is not yet set. Usually you would specify the size of the data you want to have added to each node by `NodeDataSize`, e.g. `SizeOf(TMyRecord)` is quite usual there (where `TMyRecord` is the structure you want to have stored in the node). Sometimes, however it is not possible to determine the node size in advance, so you can leave `NodeDataSize` being `-1` (the default value) and the `OnGetNodeDataSize` event is triggered as soon as the first regular node is created (the hidden root node does not have user data but internal data which is determined by other means).

See Also

`NodeDataSize`, [Data handling](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnGetNodeDataSize
Event](#) | [TVirtualStringTree.OnGetText
Event](#)

TVirtualStringTree.OnGetPopupMenu Event

[TVirtualStringTree Class](#)

Miscellaneous event.

Pascal

```
property OnGetPopupMenu: TVTPopupEvent;
```

Description

This event allows the application to return a popup menu which is specific to a certain node. The tree does an automatic traversal all the way up to the top level node which is the parent of a given node to get a popup menu. If **Menu** is set then the traversal stops. Otherwise it continues until either a menu is set, AskParent is set to False or the top level parent has been reached.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnGetPopupMenu Event](#) |
[TVirtualStringTree.OnGetUserClipboardFormats
Event](#)

TVirtualStringTree.OnGetText Event

[TVirtualStringTree Class](#) | [See Also](#)

Virtual string tree event to query for a node's normal or static text.

Pascal

```
property OnGetText: TVSTGetTextEvent;
```

Description

This is one of the fundamental string tree events which must always be handled. The string tree will fire this event every time when it needs to know about the text of a specific node and column. This is mainly the case when the node appears in the visible area of the tree view (in other words it is not scrolled out of view) but also on some other occasions, including streaming, drag and drop and calculating the width of the node.

The node text is distinguished between two text types:

- Normal text: If `TextType` is `ttNormal` return the main node caption for the specified column.
- Static text: All text that you return when `TextType` is `ttStatic` will be

displayed right beside the normal text (or left to it if the column's BidiMode is not `bdLeftToRight`, i.e. the column has right-to-left layout). Static text is used only for informational purposes; it cannot be selected or dragged and if the column is not wide enough to show all text it will not be shortened with an ellipsis (...) as normal text. The string tree will only query for static text if the `StringOptions` (see `TreeOptions`) include `toShowStaticText`. This is off by default.

When this event is fired the text parameter will always be initialized with the value of property `DefaultText`. To handle the event get your node data and then extract the string for the appropriate column and `TextType`.

Notes

Be sure that your event handler only contains absolutely necessary code. This event will be fired very often - easily a

few hundred times for medium sized trees with some columns defined when the tree is repainted completely.

For example it is far too slow to use `Locate()` on some `Dataset`, a database query result or table, and then get the text

from some `TField`. This may only work with in-memory tables or a client dataset. When you initialize your node data do

some caching and use these cached values to display the data.

See Also

OnPaintText

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class, See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnGetText](#)
[Event](#) |
[TVirtualStringTree.OnHeaderClick](#)
[Event](#)

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 Class](#) |
[TVirtualStringTree.OnGetUserClipboardFormats
Event](#) | [TVirtualStringTree.OnHeaderDbClick
Event](#)

TVirtualStringTree.OnHeaderClick Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnHeaderClick](#)
[Event](#) |
[TVirtualStringTree.OnHeaderDragged](#)
[Event](#)

TVirtualStringTree.OnHeaderDbClick Event

[TVirtualStringTree Class](#) | [See Also](#)

Header & column support event.

Pascal

```
property OnHeaderDbClick: 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 Class](#) |
[TVirtualStringTree.OnHeaderDbClick](#)
[Event](#) |
[TVirtualStringTree.OnHeaderDraggedOut](#)
[Event](#)

TVirtualStringTree.OnHeaderDragged Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnHeaderDragged](#)
[Event](#) |
[TVirtualStringTree.OnHeaderDragging](#)
[Event](#)

TVirtualStringTree.OnHeaderDraggedOut Event

[TVirtualStringTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderDraggedOut: TVTHeaderDraggedOutEvent
```

Description

When during a header drag operation the mouse moves out of the header rectangle and the mouse button is released then an OnHeaderDraggedOut event will be fired with the target mouse position in screen coordinates.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnHeaderDraggedOut
Event](#) |
[TVirtualStringTree.OnHeaderDraw Event](#)

TVirtualStringTree.OnHeaderDragging Event

[TVirtualStringTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderDragging: TVTHeaderDraggingEvent;
```

Description

Triggered just before dragging of a header button starts. Set **Allowed** to False if you want to prevent the drag operation of the given column.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnHeaderDragging Event](#) |
[TVirtualStringTree.OnHeaderDrawQueryElements
Event](#)

TVirtualStringTree.OnHeaderDraw Event

[TVirtualStringTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderDraw: TVTHeaderPaintEvent;
```

Description

If you set the hoOwnerDraw style in TVTHeader.Options and a column has been set to vsOwnerDraw (see also TVirtualTreeColumn.Style) then OnDrawHeader is called whenever a column needs painting.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnHeaderDraw Event](#) |
[TVirtualStringTree.OnHeaderMouseDown Event](#)

TVirtualStringTree.OnHeaderDrawQueryEvent

[TVirtualStringTree Class](#) | [See Also](#)

Header & column support event.

Pascal

```
property OnHeaderDrawQueryElements: TVTHeaderPaintQu
```

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 Class](#) |
[TVirtualStringTree.OnHeaderDrawQueryElements
Event](#) | [TVirtualStringTree.OnHeaderMouseMove
Event](#)

TVirtualStringTree.OnHeaderMouseDown Event

[TVirtualStringTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderMouseDown: TVHeaderMouseEvent;
```

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 Class](#) |
[TVirtualStringTree.OnHeaderMouseDown](#)
Event |
[TVirtualStringTree.OnHeaderMouseUp](#)
Event

TVirtualStringTree.OnHeaderMouseMove Event

[TVirtualStringTree Class](#)

Header & column support event.

Pascal

```
property OnHeaderMouseMove: TVHeaderMouseMoveEvent;
```

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 Class](#) |
[TVirtualStringTree.OnHeaderMouseMove
Event](#) | [TVirtualStringTree.OnHotChange
Event](#)

TVirtualStringTree.OnHeaderMouseUp Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnHeaderMouseUp](#)
[Event](#) |
[TVirtualStringTree.OnIncrementalSearch](#)
[Event](#)

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 Class](#) |
[TVirtualStringTree.OnHotChangeEvent](#) |
[TVirtualStringTree.OnInitChildrenEvent](#)

TVirtualStringTree.OnIncrementalSearchEvent

[TVirtualStringTree Class](#)

Miscellaneous event.

Pascal

```
property OnIncrementalSearch: TVTIncrementalSearchEv
```

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  
    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, No
end;

// By using StrLIComp we can specify a maximum le
// which match only partially.
Result := StrLIComp(PChar(S), PChar(PropText), Mi
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 Class](#) |
[TVirtualStringTree.OnIncrementalSearch
Event](#) | [TVirtualStringTree.OnInitNode
Event](#)

TVirtualStringTree.OnInitChildren Event

[TVirtualStringTree Class](#) | [See Also](#)

Node management event.

Pascal

```
property OnInitChildren: TVTInitChildrenEvent;
```

Description

In order to allow the tree only to fill content where needed it is possible to set the `vsHasChildren` style in a node's 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 Class](#) |
[TVirtualStringTree.OnInitChildren](#)
[Event](#) |
[TVirtualStringTree.OnKeyAction](#)
[Event](#)

TVirtualStringTree.OnInitNode Event

[TVirtualStringTree Class](#) | [See Also](#)

Node management event.

Pascal

```
property OnInitNode: TVTInitNodeEvent;
```

Description

This event is important to connect the tree to your internal data. It is the ideal place to put references or whatever you need into a node's data area. You can set some initial states like selection, expansion state or that a node has child nodes.

See Also

[The virtual paradigm](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnInitNode](#)
[Event](#) |
[TVirtualStringTree.OnKeyDown](#)
[Property](#)

TVirtualStringTree.OnKeyAction Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnKeyAction](#)
[Event](#) |
[TVirtualStringTree.OnKeyPress](#)
[Property](#)

TVirtualStringTree.OnKeyDown Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnKeyDown;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnKeyDown](#)
[Property](#) |
[TVirtualStringTree.OnKeyUp](#)
[Property](#)

TVirtualStringTree.OnKeyPress Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnKeyPress;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnKeyPress](#)
[Property](#) |
[TVirtualStringTree.OnLoadNode](#)
[Event](#)

TVirtualStringTree.OnKeyUp Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnKeyUp](#)
[Property](#) |
[TVirtualStringTree.OnMeasureItem](#)
[Event](#)

TVirtualStringTree.OnLoadNode Event

[TVirtualStringTree Class](#) | [See Also](#)

Streaming support event.

Pascal

```
property OnLoadNode: TVTSaveNodeEvent;
```

Description

This event is typically triggered when serialized tree data must be restored, e.g. when loading the tree from file or stream or during a clipboard/drag'n drop operation. You should only read in what you wrote out in `OnSaveNode`. For safety there is a check in the loader code which tries to keep the internal serialization structure intact in case the application does not read correctly.

See Also

`OnSaveNode`, `LoadFromStream`, `SaveToStream`,
`AddFromStream`, `VTTTreeStreamVersion`,
`TVTHeader.LoadFromStream`, `TVTHeader.SaveToStream`

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnLoadNode](#)
[Event](#) |
[TVirtualStringTree.OnMouseDown](#)
[Property](#)

TVirtualStringTree.OnMeasureItem Event

[TVirtualStringTree Class](#) | [See Also](#)

Miscellaneous event.

Pascal

```
property OnMeasureItem: TVTMeasureItemEvent;
```

Description

Virtual Treeview supports individual node heights. However it might sometimes be unpractical to set this height in advance (e.g. during `OnInitNode`). Another scenario might be that multi line nodes must size themselves to accommodate 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 Class](#) |
[TVirtualStringTree.OnMeasureItem](#)
[Event](#) |
[TVirtualStringTree.OnMouseMove](#)
[Property](#)

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 Class](#) |
[TVirtualStringTree.OnMouseDown](#)
[Property](#) |
[TVirtualStringTree.OnMouseUp](#)
[Property](#)

TVirtualStringTree.OnMouseMove Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnMouseMove;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnMouseMove](#)
[Property](#) |
[TVirtualStringTree.OnMouseWheel](#)
[Property](#)

TVirtualStringTree.OnMouseUp Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnMouseUp;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnMouseUp](#)
[Property](#) |
[TVirtualStringTree.OnNewText](#)
[Event](#)

TVirtualStringTree.OnMouseWheel Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnMouseWheel;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnMouseWheel](#)
[Property](#) |
[TVirtualStringTree.OnNodeCopied](#)
[Event](#)

TVirtualStringTree.OnNewText Event

[TVirtualStringTree Class](#) | [See Also](#)

Virtual string tree event to pass edited text.

Pascal

```
property OnNewText: TVSTNewTextEvent;
```

Description

A string tree will fire this event after a node has been edited successfully (not canceled with Escape). The event handler must store the new text in the node data.

This event will only be used for the default node caption editor. Other custom node editors may or may not use this event to pass their edited data to the application. Editing for the whole tree is only possible if the MiscOptions (see [TreeOptions](#)) include `toEditable`. If only certain columns or nodes should be editable write an event handler for `OnEditing`.

See Also

[OnCreateEditor](#), [OnEdited](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnNewText
Event](#) |
[TVirtualStringTree.OnNodeCopying
Event](#)

TVirtualStringTree.OnNodeCopied Event

[TVirtualStringTree Class](#)

Miscellaneous event.

Pascal

```
property OnNodeCopied: TVTNodeCopiedEvent;
```

Description

This event is triggered during drag'n drop after a node has been copied to a new location. Sender is the target tree where the copy operation took place.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnNodeCopied](#)
[Event](#) |
[TVirtualStringTree.OnNodeMoved](#)
[Event](#)

TVirtualStringTree.OnNodeCopying Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnNodeCopying](#)
[Event](#) |
[TVirtualStringTree.OnNodeMoving](#)
[Event](#)

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 Class](#) |
[TVirtualStringTree.OnNodeMoved](#)
[Event](#) |
[TVirtualStringTree.OnPaintBackground](#)
[Event](#)

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 Class](#) |
[TVirtualStringTree.OnNodeMoving](#)
[Event](#) |
[TVirtualStringTree.OnPaintText](#)
[Event](#)

TVirtualStringTree.OnPaintBackground Event

[TVirtualStringTree Class](#)

Paint support event.

Pascal

```
property OnPaintBackground: TVTBackgroundPaintEvent;
```

Description

This event is triggered when the tree has finished its painting and there is an area which is not covered by nodes. For nodes there are various events to allow background customizat0n. 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 Class](#) |
[TVirtualStringTree.OnPaintBackground](#)
[Event](#) |
[TVirtualStringTree.OnRenderOLEData](#)
[Event](#)

TVirtualStringTree.OnPaintText Event

[TVirtualStringTree Class](#) | [See Also](#)

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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnPaintText](#)
[Event](#) |
[TVirtualStringTree.OnResetNode](#)
[Event](#)

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 descendants) have specified their own formats in `OnGetUserClipboardFormats`.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnRenderOLEData
Event](#) | [TVirtualStringTree.OnResize
Property](#)

TVirtualStringTree.OnResetNode Event

[TVirtualStringTree Class](#) | [See Also](#)

Node management event.

Pascal

```
property OnResetNode: TVTChangeEvent;
```

Description

For large trees or simply because the content changed it is sometimes necessary to discard a certain node and release all its children. This can be done with `ResetNode` which will trigger this event.

See Also

`ResetNode`

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnResetNode](#)
[Event](#) |
[TVirtualStringTree.OnSaveNode](#)
[Event](#)

TVirtualStringTree.OnResize Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnResize;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnResize](#)
[Property](#) |
[TVirtualStringTree.OnScroll](#) Event

TVirtualStringTree.OnSaveNode Event

[TVirtualStringTree 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, VTTStreamVersion,
TVTHeader.LoadFromStream, TVTHeader.SaveToStream

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnSaveNode](#)
[Event](#) |
[TVirtualStringTree.OnShortenString](#)
[Event](#)

TVirtualStringTree.OnScroll Event

[TVirtualStringTree Class](#) | [See Also](#)

Miscellaneous event.

Pascal

```
property OnScroll: TVTScrollEvent;
```

Description

This event is triggered when the tree is scrolled horizontally or vertically. You can use it to synchronize scrolling of several trees or other controls.

See Also

[OffsetXY](#)

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnScroll Event](#) |
[TVirtualStringTree.OnShowScrollbar Event](#)

TVirtualStringTree.OnShortenString Event

[TVirtualStringTree Class](#)

String tree event for custom handling of string abbreviations.

Pascal

```
property OnShortenString: TVSTShortenStringEvent;
```

Description

If the text of a node does not fit into its cell (in grid mode) or is too wide for the width of the tree view it is being abbreviated with an ellipsis (...). By default the ellipsis is added to the end of the node text.

Occasionally you may want to shorten the node text at a different position, for example if the node text is a path string and not the last folder or filename should be cut off but rather some mid level folders if possible.

In the handler *S* must be processed (shortened) and returned in *Result*. If *Done* is set to true (default value is false) the tree view takes over the shortening. This is useful if not all nodes or columns need

Class

[TVirtualStringTree Class](#)

Links

TVirtualStringTree Class

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnShortenString](#)
[Event](#) |
[TVirtualStringTree.OnStartDock](#)
[Property](#)

TVirtualStringTree.OnShowScrollbar Event

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnShowScrollbar: TVTScrollbarShowEvent;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnShowScrollbar](#)
[Event](#) |
[TVirtualStringTree.OnStartDrag](#)
[Property](#)

TVirtualStringTree.OnStartDock Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property OnStartDock;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnStartDock](#)
[Property](#) |
[TVirtualStringTree.OnStateChange](#)
[Event](#)

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 Class](#) |
[TVirtualStringTree.OnStartDrag](#)
[Property](#) |
[TVirtualStringTree.OnStructureChange](#)
[Event](#)

TVirtualStringTree.OnStateChange Event

[TVirtualStringTree Class](#)

Miscellaneous event.

Pascal

```
property OnStateChange: TVTStateChangeEvent;
```

Description

For special effects or in order to increase performance it is sometimes useful to know when the tree changes one of its internal states like `tsIncrementalSearching` or `tsOLEDDragging`. 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 Class](#) |
[TVirtualStringTree.OnStateChange](#)
[Event](#) |
[TVirtualStringTree.OnUpdating](#)
[Event](#)

TVirtualStringTree.OnStructureChange Event

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnStructureChange](#)
[Event](#) |
[TVirtualStringTree.ParentBiDiMode](#)
[Property](#)

TVirtualStringTree.OnUpdating Event

[TVirtualStringTree Class](#)

Miscellaneous event.

Pascal

```
property OnUpdating: TVTUpdatingEvent;
```

Description

This event is triggered when the application or the tree call `BeginUpdate` or `EndUpdate` and indicate so when a larger update operation takes place. This can for instance be used to show a hour glass wait cursor.

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.OnUpdating](#)
[Event](#) |
[TVirtualStringTree.ParentColor](#)
[Property](#)

TVirtualStringTree.ParentBiDiMode Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property ParentBiDiMode;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.ParentBiDiMode](#)
[Property](#) |
[TVirtualStringTree.ParentCtl3D](#)
[Property](#)

TVirtualStringTree.ParentColor Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property ParentColor;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.ParentColor](#)
Property |
[TVirtualStringTree.ParentFont](#)
Property

TVirtualStringTree.ParentCtl3D Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.ParentCtl3D](#)
[Property](#) |
[TVirtualStringTree.ParentShowHint](#)
[Property](#)

TVirtualStringTree.ParentFont Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.ParentFont
Property](#) |
[TVirtualStringTree.PopupMenu
Property](#)

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 Class](#) |
[TVirtualStringTree.ParentShowHint](#)
Property |
[TVirtualStringTree.RootNodeCount](#)
Property

TVirtualStringTree.PopupMenu Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.PopupMenu
Property](#) |
[TVirtualStringTree.ScrollBarOptions
Property](#)

TVirtualStringTree.RootNodeCount Property

[TVirtualStringTree Class](#)

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

Pascal

```
property RootNodeCount: Cardinal;
```

Description

Usually setting `RootNodeCount` is all what is needed to initially fill the tree. When one of the top level nodes is initialized you can set its `ivsHasChildren` style. This will then cause to ask to initialize the child nodes. Recursively applied, you can use this principle to create tree nodes on demand (e.g. when their parent is expanded).

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.RootNodeCount](#)
[Property](#) |
[TVirtualStringTree.SelectionBlendFactor](#)
[Property](#)

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 Class](#) |
[TVirtualStringTree.ScrollBarOptions](#)
[Property](#) |
[TVirtualStringTree.SelectionCurveRadius](#)
[Property](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.SelectionBlendFactor
Property](#) | [TVirtualStringTree.ShowHint
Property](#)

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 Class](#) |
[TVirtualStringTree.SelectionCurveRadius](#)
[Property](#) |
[TVirtualStringTree.StateImages](#) [Property](#)

TVirtualStringTree.ShowHint Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property ShowHint;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.ShowHint](#)
[Property](#) |
[TVirtualStringTree.TabOrder](#)
[Property](#)

TVirtualStringTree.StateImages Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.StateImages](#)
[Property](#) |
[TVirtualStringTree.TabStop](#)
[Property](#)

TVirtualStringTree.TabOrder Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property TabOrder;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.TabOrder](#)
[Property](#) |
[TVirtualStringTree.TextMargin](#)
[Property](#)

TVirtualStringTree.TabStop Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.TabStop](#)
[Property](#) |
[TVirtualStringTree.TreeOptions](#)
[Property](#)

TVirtualStringTree.TextMargin Property

[TVirtualStringTree 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#), [See Also](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.TextMargin
Property](#) |
[TVirtualStringTree.Visible Property](#)

TVirtualStringTree.TreeOptions Property

[TVirtualStringTree Class](#)

Reference to the tree's options.

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 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

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.TreeOptions](#)
[Property](#) |
[TVirtualStringTree.WantTabs](#)
[Property](#)

TVirtualStringTree.Visible Property

[TVirtualStringTree Class](#)

Not documented.

Pascal

```
property Visible;
```

Description

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

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualStringTree Class](#) |
[TVirtualStringTree.Visible Property](#) |
[TVirtualStringTree.GetOptionsClass](#)
Method

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 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

[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

```
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(TCustomVirtualStringTr
private
    function GetOptions: TStringTreeOptions;
    procedure SetOptions(const Value: TStringTreeOp
protected
    function GetOptionsClass: TTreeOptionsClass; ov
public
    property Canvas;
published
    ...
    property TreeOptions: TStringTreeOptions read G
    ...
end;

...

//----- TVirtualStringTree -----

function TVirtualStringTree.GetOptions: TStringTree
begin
    Result := FOptions as TStringTreeOptions;
end;

//-----

procedure TVirtualStringTree.SetOptions(const Value
begin
    FOptions.Assign(Value);
end;

//-----

function TVirtualStringTree.GetOptionsClass: TTreeO
begin

```

```
Result := TStringTreeOptions;  
end;
```

Class

[TVirtualStringTree Class](#)

Links

[TVirtualStringTree Class](#)

What do you think about this topic? [Send feedback!](#)



TVirtualTreeColumn Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Represents a column in a Virtual Treeview.

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

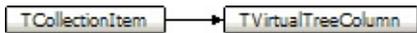


virtual



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

```
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 Class](#) |
[TVirtualTreeColumn.Alignment](#)
Property |
[TVirtualTreeColumn.Color](#)
Property

TVirtualTreeColumn.BiDiMode Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property BiDiMode: TBiDiMode;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.BiDiMode](#)
[Property](#) |
[TVirtualTreeColumn.Hint Property](#)

TVirtualTreeColumn.Color Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property Color: TColor;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.Color](#)
[Property](#) |
[TVirtualTreeColumn.ImageIndex](#)
[Property](#)

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 Class](#) |
[TVirtualTreeColumn.Hint Property](#)
| [TVirtualTreeColumn.Layout
Property](#)

TVirtualTreeColumn.ImageIndex Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property ImageIndex: TImageIndex;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.ImageIndex
Property](#) | [TVirtualTreeColumn.Left
Property](#)

TVirtualTreeColumn.Layout Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property Layout: TVTHeaderColumnLayout;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.Layout](#)
[Property](#) |
[TVirtualTreeColumn.Margin](#)
[Property](#)

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 Class](#) |
[TVirtualTreeColumn.Left Property](#) |
[TVirtualTreeColumn.MaxWidth Property](#)

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 Class](#) |
[TVirtualTreeColumn.Margin](#)
[Property](#) |
[TVirtualTreeColumn.MinWidth](#)
[Property](#)

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 Class](#) |
[TVirtualTreeColumn.MaxWidth](#)
[Property](#) |
[TVirtualTreeColumn.Options](#)
[Property](#)

TVirtualTreeColumn.MinWidth Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property MinWidth: Integer;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.MinWidth](#)
[Property](#) |
[TVirtualTreeColumn.Owner](#)
[Property](#)

TVirtualTreeColumn.Options Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property Options: TVTColumnOptions;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.Options](#)
[Property](#) |
[TVirtualTreeColumn.Position](#)
[Property](#)

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 Class](#) |
[TVirtualTreeColumn.Owner](#)
[Property](#) |
[TVirtualTreeColumn.Spacing](#)
[Property](#)

TVirtualTreeColumn.Position Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property Position: TColumnPosition;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.Position](#)
[Property](#) |
[TVirtualTreeColumn.Style Property](#)

TVirtualTreeColumn.Spacing Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property Spacing: Integer;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.Spacing](#)
[Property](#) | [TVirtualTreeColumn.Tag](#)
[Property](#)

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 Class](#) |
[TVirtualTreeColumn.Style Property](#)
| [TVirtualTreeColumn.Text
Property](#)

TVirtualTreeColumn.Tag Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property Tag: Integer;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.Tag Property](#) |
[TVirtualTreeColumn.Width Property](#)

TVirtualTreeColumn.Text Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property Text: WideString;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.Text Property](#)
| [TVirtualTreeColumn.Assign
Method](#)

TVirtualTreeColumn.Width Property

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
property Width: Integer;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.Width Property](#) |
[TVirtualTreeColumn.ComputeHeaderLayout Method](#)

TVirtualTreeColumn.Assign Method

[TVirtualTreeColumn 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

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.Assign
Method](#) |
[TVirtualTreeColumn.Create
Constructor](#)

TVirtualTreeColumn.ComputeHeaderLayo Method

[TVirtualTreeColumn Class](#)

Calculates the layout of a column header.

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 Class](#) |
[TVirtualTreeColumn.ComputeHeaderLayout](#)
Method |
[TVirtualTreeColumn.DefineProperties](#)
Method

TVirtualTreeColumn.Create Constructor

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
constructor Create(Collection: TCollection); override
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.Create](#)
[Constructor](#) |
[TVirtualTreeColumn.Destroy](#)
[Destructor](#)

TVirtualTreeColumn.DefineProperties Method

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
procedure DefineProperties(Filer: TFile); 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 Class](#) |
[TVirtualTreeColumn.DefineProperties](#)
[Method](#) | [TVirtualTreeColumn.Equals](#)
[Method](#)

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 Class](#) |
[TVirtualTreeColumn.Destroy Destructor](#) |
[TVirtualTreeColumn.GetAbsoluteBounds](#)
Method

TVirtualTreeColumn.Equals Method

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
function Equals(OtherColumn: TVirtualTreeColumn): Bo
```

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 Class](#) |
[TVirtualTreeColumn.Equals Method](#) |
[TVirtualTreeColumn.GetDisplayName Method](#)

TVirtualTreeColumn.GetAbsoluteBounds Method

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
procedure GetAbsoluteBounds(var Left: Integer; var R
```

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 Class](#) |
[TVirtualTreeColumn.GetAbsoluteBounds](#)
[Method](#) | [TVirtualTreeColumn.GetOwner](#)
[Method](#)

TVirtualTreeColumn.GetDisplayName Method

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
function GetDisplayName: string; override;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.GetDisplayName](#)
Method |
[TVirtualTreeColumn.GetRect](#) Method

TVirtualTreeColumn.GetOwner Method

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
function GetOwner: TVirtualTreeColumns; reintroduce;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.GetOwner](#)
Method |
[TVirtualTreeColumn.LoadFromStream](#)
Method

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 Class](#) |
[TVirtualTreeColumn.GetRect Method](#) |
[TVirtualTreeColumn.ParentBiDiModeChanged Method](#)

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 Class](#) |
[TVirtualTreeColumn.LoadFromStream](#)
Method |
[TVirtualTreeColumn.ParentColorChanged](#)
Method

TVirtualTreeColumn.ParentBiDiModeChar 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 Class](#) |
[TVirtualTreeColumn.ParentBiDiModeChanged
Method](#) | [TVirtualTreeColumn.ReadHint
Method](#)

TVirtualTreeColumn.ParentColorChanged Method

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
procedure ParentColorChanged;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.ParentColorChanged
Method](#) | [TVirtualTreeColumn.ReadText
Method](#)

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 Class](#) |
[TVirtualTreeColumn.ReadHint Method](#) |
[TVirtualTreeColumn.RestoreLastWidth Method](#)

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 Class](#) |
[TVirtualTreeColumn.ReadText](#)
Method |
[TVirtualTreeColumn.SaveToStream](#)
Method

TVirtualTreeColumn.RestoreLastWidth Method

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
procedure RestoreLastWidth;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.RestoreLastWidth](#)
[Method](#) |
[TVirtualTreeColumn.UseRightToLeftReading](#)
[Method](#)

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 Class](#) |
[TVirtualTreeColumn.SaveToStream](#)
Method |
[TVirtualTreeColumn.WriteHint](#)
Method

TVirtualTreeColumn.UseRightToLeftReading Method

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
function UseRightToLeftReading: Boolean;
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumn Class](#) |
[TVirtualTreeColumn.UseRightToLeftReading
Method](#) | [TVirtualTreeColumn.WriteText
Method](#)

TVirtualTreeColumn.WriteHint Method

[TVirtualTreeColumn Class](#)

Not documented.

Pascal

```
procedure WriteHint(Writer: TWriter);
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



TVirtualTreeColumn.WriteText Method

TVirtualTreeColumn Class

Not documented.

Pascal

```
procedure WriteText(Writer: TWriter);
```

Description

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

Class

[TVirtualTreeColumn Class](#)

Links

[TVirtualTreeColumn Class](#)

What do you think about this topic? [Send feedback!](#)



TVirtualTreeColumns Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Collection class, which holds the columns for the tree.

Pascal

```
TVirtualTreeColumns = class(TCollection);
```

Description

This class is an enhanced collection which manages general aspects of columns like ordering, 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](#)

Not documented.

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.

  **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 PositionToIndex array, which primarily determines where each column is placed visually.

Legend



public



Property



read only



protected

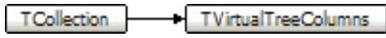


Method



virtual

Class Hierarchy



File

VirtualTrees

Links

[Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TVirtualTreeColumns.ClickIndex Property

TVirtualTreeColumns Class

Not documented.

Pascal

```
property ClickIndex: TColumnIndex;
```

Description

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

Class

TVirtualTreeColumns Class

Links

TVirtualTreeColumns Class

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.ClickIndex](#)
[Property](#) |
[TVirtualTreeColumns.HeaderBitmap](#)
[Property](#)

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 Class](#) |
[TVirtualTreeColumns.Header](#)
[Property](#) |
[TVirtualTreeColumns.Items](#)
[Property](#)

TVirtualTreeColumns.HeaderBitmap Property

[TVirtualTreeColumns Class](#)

Not documented.

Pascal

```
property HeaderBitmap: TBitmap;
```

Description

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

Class

[TVirtualTreeColumns Class](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.HeaderBitmap](#)
[Property](#) |
[TVirtualTreeColumns.PositionToIndex](#)
[Property](#)

TVirtualTreeColumns.Items Property

[TVirtualTreeColumns Class](#)

Not documented.

Pascal

```
property Items [Index: TColumnIndex]: TVirtualTreeCo
```

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 Class](#) |
[TVirtualTreeColumns.Items](#)
[Property](#) |
[TVirtualTreeColumns.TrackIndex](#)
[Property](#)

TVirtualTreeColumns.PositionToIndex Property

[TVirtualTreeColumns Class](#)

Not documented.

Pascal

```
property PositionToIndex: TIndexArray;
```

Description

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

Class

[TVirtualTreeColumns Class](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.PositionToIndex
Property](#) | [TVirtualTreeColumns.Add
Method](#)

TVirtualTreeColumns.TrackIndex Property

[TVirtualTreeColumns Class](#)

Not documented.

Pascal

```
property TrackIndex: TColumnIndex;
```

Description

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

Class

[TVirtualTreeColumns Class](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.TrackIndex](#)
[Property](#) |
[TVirtualTreeColumns.AdjustAutoSize](#)
[Method](#)

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 Class](#) |
[TVirtualTreeColumns.Add Method](#) |
[TVirtualTreeColumns.AdjustDownColumn Method](#)

TVirtualTreeColumns.AdjustAutoSize Method

[TVirtualTreeColumns Class](#)

Called when columns must be sized so that they fit the client area.

Pascal

```
procedure AdjustAutoSize(CurrentIndex: TColumnIndex;
```

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.AdjustAutoSize](#)
Method |
[TVirtualTreeColumns.AdjustHoverColumn](#)
Method

TVirtualTreeColumns.AdjustDownColumn Method

[TVirtualTreeColumns Class](#)

Determines the column from the given position and returns it.

Pascal

```
function AdjustDownColumn(P: TPoint): TColumnIndex;
```

Description

If this column is allowed to be clicked then it is also kept for later use.

Class

[TVirtualTreeColumns Class](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.AdjustDownColumn](#)
Method |
[TVirtualTreeColumns.AdjustPosition](#)
Method

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.AdjustHoverColumn](#)
Method |
[TVirtualTreeColumns.AnimatedResize](#)
Method

TVirtualTreeColumns.AdjustPosition Method

[TVirtualTreeColumns Class](#)

Reorders the column position array so that the given column gets the given position.

Pascal

```
procedure AdjustPosition(Column: TVirtualTreeColumn;
```

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 Class](#) |
[TVirtualTreeColumns.AdjustPosition](#)
Method |
[TVirtualTreeColumns.Assign](#)
Method

TVirtualTreeColumns.AnimatedResize Method

[TVirtualTreeColumns Class](#)

Resizes the given column animated by scrolling the window DC.

Pascal

```
procedure AnimatedResize(Column: TColumnIndex; NewWi
```

Description

Resizes the given column animated by scrolling the window DC.

Class

[TVirtualTreeColumns Class](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.AnimatedResize
Method](#) | [TVirtualTreeColumns.Clear
Method](#)

TVirtualTreeColumns.Assign Method

[TVirtualTreeColumns 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

[TVirtualTreeColumns Class](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.Assign
Method](#) |
[TVirtualTreeColumns.Create
Constructor](#)

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 Class |
TVirtualTreeColumns.ColumnFromPosition
Method (TPoint, Boolean)

TVirtualTreeColumns.ColumnFromPosition Method (TColumnPosition)

TVirtualTreeColumns Class

Returns the index of the column at the given position.

Pascal

```
function ColumnFromPosition(PositionIndex: TColumnPo
```

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 Class |
TVirtualTreeColumns.ColumnFromPosition
Method (TColumnPosition)

TVirtualTreeColumns.ColumnFromPosition Method (TPoint, Boolean)

TVirtualTreeColumns Class

Determines the current column based on the position passed in P.

Pascal

```
function ColumnFromPosition(P: TPoint; Relative: Boo
```

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 Class](#) |
[TVirtualTreeColumns.Clear](#)
Method |
[TVirtualTreeColumns.Destroy](#)
Destructor

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 Class](#) |
[TVirtualTreeColumns.Create](#)
[Constructor](#) |
[TVirtualTreeColumns.DrawButtonText](#)
[Method](#)

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 Class](#) |
[TVirtualTreeColumns.Destroy](#)
[Destructor](#) |
[TVirtualTreeColumns.DrawXPButton](#)
[Method](#)

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 Class](#) |
[TVirtualTreeColumns.DrawButtonText](#)
[Method](#) |
[TVirtualTreeColumns.Equals Method](#)

TVirtualTreeColumns.DrawXPButton Method

[TVirtualTreeColumns Class](#)

Helper procedure to draw an Windows XP like header button.

Pascal

```
procedure DrawXPButton(DC: HDC; ButtonR: TRect; Draw
```

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 Class](#) |
[TVirtualTreeColumns.DrawXPButton](#)
[Method](#) |
[TVirtualTreeColumns.FixPositions](#)
[Method](#)

TVirtualTreeColumns.Equals Method

[TVirtualTreeColumns Class](#)

Compares itself with the given set of columns.

Pascal

```
function Equals(OtherColumns: TVirtualTreeColumns):
```

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 Class](#) |
[TVirtualTreeColumns.Equals Method](#) |
[TVirtualTreeColumns.GetColumnAndBounds Method](#)

TVirtualTreeColumns.FixPositions Method

[TVirtualTreeColumns Class](#)

Fixes column positions after loading from DFM.

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 Class](#) |
[TVirtualTreeColumns.FixPositions](#)
[Method](#) |
[TVirtualTreeColumns.GetColumnBounds](#)
[Method](#)

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: Integer; var 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 Class](#) |
[TVirtualTreeColumns.GetColumnAndBounds](#)
Method |
[TVirtualTreeColumns.GetFirstVisibleColumn](#)
Method

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 Class](#) |
[TVirtualTreeColumns.GetColumnBounds](#)
Method |
[TVirtualTreeColumns.GetLastVisibleColumn](#)
Method

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.GetFirstVisibleColumn](#)
Method |
[TVirtualTreeColumns.GetNextColumn](#)
Method

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 Class](#) |
[TVirtualTreeColumns.GetLastVisibleColumn](#)
Method |
[TVirtualTreeColumns.GetNextVisibleColumn](#)
Method

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): TColumn
```

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.GetNextColumn](#)
Method |
[TVirtualTreeColumns.GetOwner](#)
Method

TVirtualTreeColumns.GetNextVisibleColumn Method

[TVirtualTreeColumns Class](#)

Returns the next visible column in display order, Column is an index into the columns list.

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 Class](#) |
[TVirtualTreeColumns.GetNextVisibleColumn](#)
Method |
[TVirtualTreeColumns.GetPreviousColumn](#)
Method

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 Class](#) |
[TVirtualTreeColumns.GetOwner Method](#) |
[TVirtualTreeColumns.GetPreviousVisibleColumn Method](#)

TVirtualTreeColumns.GetPreviousColumn Method

[TVirtualTreeColumns Class](#)

Returns the previous column in display order, Column is an index into the columns list.

Pascal

```
function GetPreviousColumn(Column: TColumnIndex): TC
```

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 Class](#) |
[TVirtualTreeColumns.GetPreviousColumn](#)
[Method](#) |
[TVirtualTreeColumns.GetVisibleColumns](#)
[Method](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.GetPreviousVisibleColumn](#)
Method |
[TVirtualTreeColumns.GetVisibleFixedWidth](#)
Method

TVirtualTreeColumns.GetVisibleColumns Method

[TVirtualTreeColumns Class](#)

Returns a list of all currently visible columns in actual order.

Pascal

```
function GetVisibleColumns: TColumnsArray;
```

Description

Returns a list of all currently visible columns in actual order.

Class

[TVirtualTreeColumns Class](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.GetVisibleColumns](#)
Method |
[TVirtualTreeColumns.HandleClick](#)
Method

TVirtualTreeColumns.GetVisibleFixedWidth Method

[TVirtualTreeColumns Class](#)

Not documented.

Pascal

```
function GetVisibleFixedWidth: Integer;
```

Description

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

Class

[TVirtualTreeColumns Class](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.GetVisibleFixedWidth](#)
[Method](#) |
[TVirtualTreeColumns.IndexChanged](#)
[Method](#)

TVirtualTreeColumns.HandleClick Method

[TVirtualTreeColumns Class](#)

Generates a click event if the mouse button has been released over the same column it was pressed first.

Pascal

```
procedure HandleClick(P: TPoint; Button: TMouseButton
```

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](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.HandleClick Method](#) |
[TVirtualTreeColumns.InitializePositionArray Method](#)

TVirtualTreeColumns.IndexChanged Method

[TVirtualTreeColumns Class](#)

Called by a column when its index in the collection changes.

Pascal

```
procedure IndexChanged(OldIndex: Integer; NewIndex:
```

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 Class](#) |
[TVirtualTreeColumns.IndexChanged](#)
Method |
[TVirtualTreeColumns.IsValidColumn](#)
Method

TVirtualTreeColumns.InitializePositionArr Method

[TVirtualTreeColumns Class](#)

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 Class](#) |
[TVirtualTreeColumns.InitializePositionArray](#)
Method |
[TVirtualTreeColumns.LoadFromStream](#)
Method

TVirtualTreeColumns.IsValidColumn Method

[TVirtualTreeColumns Class](#)

Determines whether the given column is valid or not, that is, whether it is one of the current columns.

Pascal

```
function IsValidColumn(Column: TColumnIndex): Boolean;
```

Description

Determines whether the given column is valid or not, that is, whether it is one of the current columns.

Class

[TVirtualTreeColumns Class](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.IsValidColumn](#)
[Method](#) |
[TVirtualTreeColumns.PaintHeader](#)
[Method](#)

TVirtualTreeColumns.LoadFromStream Method

[TVirtualTreeColumns 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

[TVirtualTreeColumns Class](#)

Links

[TVirtualTreeColumns Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeColumns Class](#) |
[TVirtualTreeColumns.LoadFromStream](#)
Method |
[TVirtualTreeColumns.SaveToStream](#)
Method

TVirtualTreeColumns.PaintHeader Method

[TVirtualTreeColumns Class](#)

Not documented.

Pascal

```
procedure PaintHeader(DC: HDC; R: TRect; HOffset: In
```

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 Class](#) |
[TVirtualTreeColumns.PaintHeader](#)
[Method](#) |
[TVirtualTreeColumns.TotalWidth](#)
[Method](#)

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 Class](#) |
[TVirtualTreeColumns.SaveToStream](#)
Method |
[TVirtualTreeColumns.Update](#)
Method

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 Class](#) |
[TVirtualTreeColumns.TotalWidth](#)
Method |
[TVirtualTreeColumns.UpdatePositions](#)
Method

TVirtualTreeColumns.Update Method

[TVirtualTreeColumns Class](#)

Not documented.

Pascal

```
procedure Update(Item: TCollectionItem); override;
```

Description

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

Class

[TVirtualTreeColumns Class](#)

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 PositionToIndex 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

[Classes](#) | [Methods](#) | [Legend](#)

Internally used hint window class to support Unicode hints.

Pascal

```
TVirtualTreeHintWindow = class(THintWindow);
```

Description

TVirtualTreeHintWindow replaces Delphi's own hint window, but only for the tree controls. For the rest of the application the hint stays at it is. This means not the global HintWindowClass variable is changed but only the locally used class by properly responding to CM_HINTSHOW.

Group

[Classes](#)

Members

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



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: str
```

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 Class](#) |
[TVirtualTreeHintWindow.ActivateHint](#)
[Method](#) |
[TVirtualTreeHintWindow.Create](#)
[Constructor](#)

TVirtualTreeHintWindow.CalcHintRect Method

[TVirtualTreeHintWindow Class](#)

Not documented.

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 Class](#) |
[TVirtualTreeHintWindow.CalcHintRect](#)
[Method](#) |
[TVirtualTreeHintWindow.CreateParams](#)
[Method](#)

TVirtualTreeHintWindow.Create Constructor

[TVirtualTreeHintWindow Class](#)

Not documented.

Pascal

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

Description

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

Class

[TVirtualTreeHintWindow Class](#)

Links

[TVirtualTreeHintWindow Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeHintWindow Class](#) |
[TVirtualTreeHintWindow.Create](#)
[Constructor](#) |
[TVirtualTreeHintWindow.Destroy](#)
[Destructor](#)

TVirtualTreeHintWindow.CreateParams Method

[TVirtualTreeHintWindow Class](#)

Not documented.

Pascal

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

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 Class](#) |
[TVirtualTreeHintWindow.CreateParams](#)
Method |
[TVirtualTreeHintWindow.IsHintMsg](#)
Method

TVirtualTreeHintWindow.Destroy Destructor

[TVirtualTreeHintWindow 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

[TVirtualTreeHintWindow Class](#)

Links

[TVirtualTreeHintWindow Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeHintWindow Class](#) |
[TVirtualTreeHintWindow.Destroy](#)
[Destructor](#) |
[TVirtualTreeHintWindow.Paint](#)
[Method](#)

TVirtualTreeHintWindow.IsHintMsg Method

[TVirtualTreeHintWindow Class](#)

The VCL is a bit too generous when telling that an existing hint can be cancelled.

Pascal

```
function IsHintMsg(var Msg: TMsg): Boolean; override
```

Description

Need to specify further here.

Class

[TVirtualTreeHintWindow Class](#)

Links

[TVirtualTreeHintWindow Class](#)

What do you think about this topic? [Send feedback!](#)



TVirtualTreeHintWindow.Paint Method

TVirtualTreeHintWindow Class

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!](#)



[Classes](#) | [TVirtualTreeHintWindow Class](#) | [TVTColors Class](#)

TVirtualTreeOptions Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

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

Pascal

```
TVirtualTreeOptions = class(TCustomVirtualTreeOption
```

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

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeOptions Class](#) |
[TVirtualTreeOptions.AnimationOptions](#)
[Property](#) |
[TVirtualTreeOptions.MiscOptions](#)
[Property](#)

TVirtualTreeOptions.AutoOptions Property

[TVirtualTreeOptions Class](#)

Options related to automatic actions.

Pascal

```
property AutoOptions: TVTAutoOptions;
```

Description

These options can be used to switch certain actions in a tree which happen automatically under certain circumstances.

Class

[TVirtualTreeOptions Class](#)

Links

[TVirtualTreeOptions Class](#)

What do you think about this topic? [Send feedback!](#)



[TVirtualTreeOptions Class](#) |
[TVirtualTreeOptions.AutoOptions](#)
[Property](#) |
[TVirtualTreeOptions.PaintOptions](#)
[Property](#)

TVirtualTreeOptions.MiscOptions Property

[TVirtualTreeOptions Class](#)

Options not related to any other category.

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 Class](#) |
[TVirtualTreeOptions.MiscOptions](#)
[Property](#) |
[TVirtualTreeOptions.SelectionOptions](#)
[Property](#)

TVirtualTreeOptions.PaintOptions Property

[TVirtualTreeOptions Class](#)

Options related to painting.

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

```
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!](#)



[Classes](#) | [TVirtualTreeOptions Class](#) | [TVTDataObject Class](#)

TVTColors Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Collects all color related options for the tree control.

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

```
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 Class](#) |
[TVTColors.BorderColor Property](#) |
[TVTColors.DropMarkColor Property](#)

TVTColors.DisabledColor Property

[TVTColors Class](#)

Not documented.

Pascal

```
property DisabledColor: TColor;
```

Description

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

Class

[TVTColors Class](#)

Links

[TVTColors Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTColors Class](#) |
[TVTColors.DisabledColor Property](#) |
[TVTColors.DropTargetBorderColor Property](#)

TVTColors.DropMarkColor Property

[TVTColors Class](#)

Color of the drop mark.

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

[TVTColors Class](#)

Links

[TVTColors Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTColors Class](#) |
[TVTColors.DropMarkColor](#)
[Property](#) |
[TVTColors.DropTargetColor](#)
[Property](#)

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 Class](#) |
[TVTColors.DropTargetBorderColor](#)
[Property](#) |
[TVTColors.FocusedSelectionBorderColor](#)
[Property](#)

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 Class](#) |
[TVTColors.DropTargetColor](#)
[Property](#) |
[TVTColors.FocusedSelectionColor](#)
[Property](#)

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 Class |
TVTColors.FocusedSelectionBorderColor
Property | TVTColors.GridLineColor
Property

TVTColors.FocusedSelectionColor Property

TVTColors Class

Not documented.

Pascal

```
property FocusedSelectionColor: TColor;
```

Description

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

Class

TVTColors Class

Links

TVTColors Class

What do you think about this topic? [Send feedback!](#)



[TVTColors Class](#) |
[TVTColors.FocusedSelectionColor](#)
[Property](#) |
[TVTColors.HeaderHotColor](#)
[Property](#)

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 Class](#) |
[TVTColors.GridLineColor Property](#)
| [TVTColors.HotColor Property](#)

TVTColors.HeaderHotColor Property

[TVTColors Class](#)

Not documented.

Pascal

```
property HeaderHotColor: TColor;
```

Description

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

Class

[TVTColors Class](#)

Links

[TVTColors Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTColors Class](#) |
[TVTColors.HeaderHotColor Property](#) |
[TVTColors.SelectionRectangleBlendColor Property](#)

TVTColors.HotColor Property

[TVTColors Class](#)

Not documented.

Pascal

```
property HotColor: TColor;
```

Description

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

Class

[TVTColors Class](#)

Links

[TVTColors Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTColors Class](#) | [TVTColors.HotColor Property](#) | [TVTColors.SelectionRectangleBorderColor Property](#)

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 Class](#) |
[TVTColors.SelectionRectangleBlendColor
Property](#) | [TVTColors.TreeLineColor
Property](#)

TVTColors.SelectionRectangleBorderColor Property

[TVTColors Class](#)

Not documented.

Pascal

```
property SelectionRectangleBorderColor: TColor;
```

Description

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

Class

[TVTColors Class](#)

Links

[TVTColors Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTColors Class](#) |
[TVTColors.SelectionRectangleBorderColor](#)
[Property](#) |
[TVTColors.UnfocusedSelectionBorderColor](#)
[Property](#)

TVTColors.TreeLineColor Property

[TVTColors Class](#)

Not documented.

Pascal

```
property TreeLineColor: TColor;
```

Description

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

Class

[TVTColors Class](#)

Links

[TVTColors Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTColors Class](#) |
[TVTColors.TreeLineColor Property](#) |
[TVTColors.UnfocusedSelectionColor Property](#)

TVTColors.UnfocusedSelectionBorderColor Property

[TVTColors Class](#)

Not documented.

Pascal

```
property UnfocusedSelectionBorderColor: TColor;
```

Description

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

Class

[TVTColors Class](#)

Links

[TVTColors Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTColors Class](#) |
[TVTColors.UnfocusedSelectionBorderColor
Property](#) | [TVTColors.Assign Method](#)

TVTColors.UnfocusedSelectionColor Property

[TVTColors Class](#)

Not documented.

Pascal

```
property UnfocusedSelectionColor: TColor;
```

Description

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

Class

[TVTColors Class](#)

Links

[TVTColors Class](#)

What do you think about this topic? [Send feedback!](#)



TVTColors Class |
TVTColors.UnfocusedSelectionColor
Property | TVTColors.Create
Constructor

TVTColors.Assign Method

TVTColors 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

TVTColors Class

Links

TVTColors Class

What do you think about this topic? [Send feedback!](#)



TVTColors.Create Constructor

TVTColors Class

Not documented.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree);
```

Description

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

Class

TVTColors Class

Links

TVTColors Class

What do you think about this topic? [Send feedback!](#)



[Classes](#) | [TVTColors Class](#) | [TVTDragImage Class](#)

TVTDataObject Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Implementation of an IDataObject interface.

Pascal

```
TVTDataObject = class(TInterfacedObject, IDataObject)
```

Description

This class is used for OLE drag'n drop and clipboard operations. It allows not only to transfer various kinds of data between trees but also to transfer this data between different processes. Additionally, every OLE aware application (like Word) can take part in the data transfer. This makes it easy to copy some of the tree's content for documentation purposes.

Group

[Classes](#)

Members

Properties

-  [ForClipboard](#)
Not documented.
-  [FormatEtcArray](#)
Not documented.
-  [InternalStgMediumArray](#)
Not documented.
-  [Owner](#)
Not documented.

Methods

-  **CanonicalUnknown**
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**
Searchs 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.

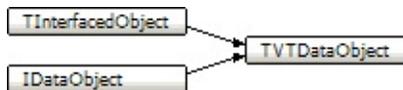
 [StgMediumIncRef](#)

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

```
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 Class](#) |
[TVTDataObject.ForClipboard Property](#) |
[TVTDataObject.InternalStgMediumArray](#)
[Property](#)

TVTDataObject.FormatEtcArray Property

[TVTDataObject Class](#)

Not documented.

Pascal

```
property FormatEtcArray: TFormatEtcArray;
```

Description

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

Class

[TVTDataObject Class](#)

Links

[TVTDataObject Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDataObject Class](#) |
[TVTDataObject.FormatEtcArray
Property](#) | [TVTDataObject.Owner
Property](#)

TVTDataObject.InternalStgMediumArray Property

[TVTDataObject Class](#)

Not documented.

Pascal

```
property InternalStgMediumArray: TInternalStgMediumA
```

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 Class](#) |
[TVTDataObject.InternalStgMediumArray](#)
[Property](#) |
[TVTDataObject.CanonicalUnknown](#)
[Method](#)

TVTDataObject.Owner Property

[TVTDataObject Class](#)

Not documented.

Pascal

```
property Owner: TBaseVirtualTree;
```

Description

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

Class

[TVTDataObject Class](#)

Links

[TVTDataObject Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDataObject Class](#) |
[TVTDataObject.Owner Property](#) |
[TVTDataObject.Create
Constructor](#)

TVTDataObject.CanonicalUnknown Method

[TVTDataObject Class](#)

Helper method for setting data in the IDataObject.

Pascal

```
function CanonicalUnknown(TestUnknown: IUnknown): I
```

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. CanonicalUnknown 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](#)

What do you think about this topic? [Send feedback!](#)



[TVTDataObject Class](#) |
[TVTDataObject.CanonicalUnknown](#)
[Method](#) | [TVTDataObject.DAdvise](#)
[Method](#)

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 Class](#) |
[TVTDataObject.Create](#)
[Constructor](#) |
[TVTDataObject.Destroy](#) [Destructor](#)

TVTDataObject.DAdvise Method

[TVTDataObject Class](#)

Implementation of the IDataObject.DAdvise method.

Pascal

```
function DAdvise(const FormatEtc: TFormatEtc; advf:
```

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 Class](#) |
[TVTDataObject.DAdvise Method](#) |
[TVTDataObject.DUnadvise Method](#)

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 Class](#) |
[TVTDataObject.Destroy Destructor](#)
| [TVTDataObject.EnumDAdvise](#)
[Method](#)

TVTDataObject.DUnadvise Method

[TVTDataObject Class](#)

Implementation of the IDataObject.DUnAdvise method.

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!](#)



[TVTDataObject Class](#) |
[TVTDataObject.DUnadvise](#)
[Method](#) |
[TVTDataObject.EnumFormatEtc](#)
[Method](#)

TVTDataObject.EnumDAdvise Method

[TVTDataObject Class](#)

Implementation of the IDataObject.EnumDAdvise method.

Pascal

```
function EnumDAdvise(out enumAdvise: IEnumStatData):
```

Description

EnumDAdvise does nothing but forwards the call to the internal advise holder class, which then responds accordingly. That's why we use data advise holders after all.

Class

[TVTDataObject Class](#)

Links

[TVTDataObject Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDataObject Class](#) |
[TVTDataObject.EnumDAdvise](#)
[Method](#) |
[TVTDataObject.EqualFormatEtc](#)
[Method](#)

TVTDataObject.EnumFormatEtc Method

[TVTDataObject Class](#)

Implementation of the IDataObject.EnumFormatEtc method.

Pascal

```
function EnumFormatEtc(Direction: Integer; out EnumF
```

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 Class](#) |
[TVTDataObject.EnumFormatEtc](#)
[Method](#) |
[TVTDataObject.FindFormatEtc](#)
[Method](#)

TVTDataObject.EqualFormatEtc Method

[TVTDataObject Class](#)

Compares two TFormatEtc structures.

Pascal

```
function EqualFormatEtc(FormatEtc1: TFormatEtc; Form
```

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 Class](#) |
[TVTDataObject.EqualFormatEtc](#)
[Method](#) |
[TVTDataObject.FindInternalStgMedium](#)
[Method](#)

TVTDataObject.FindFormatEtc Method

[TVTDataObject Class](#)

Searchs the given array for a the given format.

Pascal

```
function FindFormatEtc(TestFormatEtc: TFormatEtc; co
```

Description

Returns true if the given format is part of the array.

Class

[TVTDataObject Class](#)

Links

[TVTDataObject Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDataObject Class](#) |
[TVTDataObject.FindFormatEtc Method](#) |
[TVTDataObject.GetCanonicalFormatEtc Method](#)

TVTDataObject.FindInternalStgMedium Method

[TVTDataObject Class](#)

Returns a storage medium for a given clipboard format.

Pascal

```
function FindInternalStgMedium(Format: TClipFormat):
```

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 Class](#) |
[TVTDataObject.FindInternalStgMedium
Method](#) | [TVTDataObject.GetData
Method](#)

TVTDataObject.GetCanonicalFormatEtc Method

[TVTDataObject Class](#)

Implementation of the IDataObject.GetCanonicalFormatEtc method.

Pascal

```
function GetCanonicalFormatEtc(const FormatEtc: TFor
```

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 Class](#) |
[TVTDataObject.GetCanonicalFormatEtc](#)
[Method](#) | [TVTDataObject.GetDataHere](#)
[Method](#)

TVTDataObject.GetData Method

[TVTDataObject Class](#) | [See Also](#)

Implementation of the `IDataObject.GetData` method.

Pascal

```
function GetData(const FormatEtcIn: TFormatEtc; out
```

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 Class](#) |
[TVTDataObject.GetData Method](#) |
[TVTDataObject.HGlobalClone Method](#)

TVTDataObject.GetDataHere Method

[TVTDataObject Class](#)

Implementation of the IDataObject.GetDataHere method.

Pascal

```
function GetDataHere(const FormatEtc: TFormatEtc; ou
```

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 Class](#) |
[TVTDataObject.GetDataHere](#)
Method |
[TVTDataObject.QueryGetData](#)
Method

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 Class |
TVTDataObject.HGlobalClone Method |
TVTDataObject.RenderInternalOLEData
Method

TVTDataObject.QueryGetData Method

TVTDataObject Class

Implementation of the IDataObject.QueryGetData method.

Pascal

```
function QueryGetData(const FormatEtc: TFormatEtc):
```

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 Class](#) |
[TVTDataObject.QueryGetData](#)
[Method](#) | [TVTDataObject.SetData](#)
[Method](#)

TVTDataObject.RenderInternalOLEData Method

[TVTDataObject Class](#)

Helper method to return data previously stored by [SetData](#).

Pascal

```
function RenderInternalOLEData(const FormatEtcIn: TF
```

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 Class](#) |
[TVTDataObject.RenderInternalOLEData](#)
[Method](#) |
[TVTDataObject.StgMediumIncRef](#)
[Method](#)

TVTDataObject.SetData Method

[TVTDataObject Class](#)

Implementation of the IDataObject.SetData method.

Pascal

```
function SetData(const FormatEtc: TFormatEtc; var Me
```

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

```
function StgMediumIncRef(const InStgMedium: TStgMedi
```

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!](#)



[Classes](#) | [TVTDataObject Class](#) |
[TVTDragManager Class](#)

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

`TVTDragImage`

File

VirtualTrees

Links

[Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TVTDragImage.ColorKey Property

TVTDragImage Class

Not documented.

Pascal

```
property ColorKey: TColor;
```

Description

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

Class

[TVTDragImage Class](#)

Links

[TVTDragImage Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDragImage Class](#) |
[TVTDragImage.ColorKey Property](#)
| [TVTDragImage.MoveRestriction Property](#)

TVTDragImage.Fade Property

[TVTDragImage Class](#)

Not documented.

Pascal

```
property Fade: Boolean;
```

Description

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

Class

[TVTDragImage Class](#)

Links

[TVTDragImage Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDragImage Class](#) |
[TVTDragImage.Fade Property](#) |
[TVTDragImage.PostBlendBias Property](#)

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 Class](#) |
[TVTDragImage.MoveRestriction](#)
[Property](#) |
[TVTDragImage.PreBlendBias](#)
[Property](#)

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 Class](#) |
[TVTDragImage.PostBlendBias](#)
[Property](#) |
[TVTDragImage.Transparency](#)
[Property](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVTDragImage Class](#) |
[TVTDragImage.PreBlendBias
Property](#) | [TVTDragImage.Visible
Property](#)

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 Class](#) |
[TVTDragImage.Transparency](#)
[Property](#) | [TVTDragImage.Create](#)
[Constructor](#)

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 Class](#) |
[TVTDragImage.Visible Property](#) |
[TVTDragImage.Destroy Destructor](#)

TVTDragImage.Create Constructor

[TVTDragImage Class](#)

Not documented.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree);
```

Description

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

Class

[TVTDragImage Class](#)

Links

[TVTDragImage Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDragImage Class](#) |
[TVTDragImage.Create](#)
[Constructor](#) |
[TVTDragImage.DragTo](#) Method

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 Class](#) |
[TVTDragImage.Destroy Destructor](#)
| [TVTDragImage.EndDrag Method](#)

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): B
```

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 Class](#) |
[TVTDragImage.DragTo Method](#) |
[TVTDragImage.GetDragImageRect Method](#)

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 Class](#) |
[TVTDragImage.EndDrag Method](#) |
[TVTDragImage.HideDragImage Method](#)

TVTDragImage.GetDragImageRect Method

[TVTDragImage Class](#)

Returns the current size and position of the drag image (screen coordinates).

Pascal

```
function GetDragImageRect: TRect;
```

Description

Returns the current size and position of the drag image (screen coordinates).

Class

[TVTDragImage Class](#)

Links

[TVTDragImage Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDragImage Class](#) |
[TVTDragImage.GetDragImageRect](#)
Method |
[TVTDragImage.InternalShowDragImage](#)
Method

TVTDragImage.HideDragImage Method

[TVTDragImage Class](#)

Not documented.

Pascal

```
procedure HideDragImage;
```

Description

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

Class

[TVTDragImage Class](#)

Links

[TVTDragImage Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDragImage Class](#) |
[TVTDragImage.HideDragImage](#)
Method |
[TVTDragImage.MakeAlphaChannel](#)
Method

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 Class](#) |
[TVTDragImage.InternalShowDragImage
Method](#) | [TVTDragImage.PrepareDrag
Method](#)

TVTDragImage.MakeAlphaChannel Method

[TVTDragImage Class](#)

Not documented.

Pascal

```
procedure MakeAlphaChannel(Source: TBitmap; Target:
```

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 Class](#) |
[TVTDragImage.MakeAlphaChannel](#)
[Method](#) |
[TVTDragImage.RecaptureBackground](#)
[Method](#)

TVTDragImage.PrepareDrag Method

[TVTDragImage Class](#)

Creates all necessary structures to do alpha blended dragging using the given image.

Pascal

```
procedure PrepareDrag(DragImage: TBitmap; ImagePosit
```

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 Class](#) |
[TVTDragImage.PrepareDrag](#)
Method |
[TVTDragImage.ShowDragImage](#)
Method

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)
```

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 Class](#) |
[TVTDragImage.RecaptureBackground
Method](#) | [TVTDragImage.WillMove
Method](#)

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

```
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

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Not documented.

Pascal

```
TVTDragManager = class(TInterfacedObject, IVTDragMan
```

Description

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

Group

[Classes](#)

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**

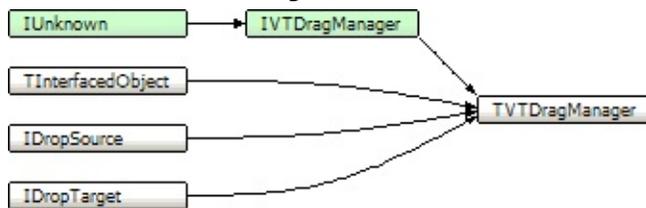
Not documented.

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 Class](#) |
[TVTDragManager.Destroy](#)
[Destructor](#)

TVTDragManager.Create Constructor

[TVTDragManager Class](#)

Not documented.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree); virtual
```

Description

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

Class

[TVTDragManager Class](#)

Links

[TVTDragManager Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDragManager Class](#) |
[TVTDragManager.Create](#)
[Constructor](#) |
[TVTDragManager.DragEnter](#)
[Method](#)

TVTDragManager.Destroy 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 Class](#) |
[TVTDragManager.Destroy](#)
[Destructor](#) |
[TVTDragManager.DragLeave](#)
[Method](#)

TVTDragManager.DragEnter Method

[TVTDragManager Class](#)

Not documented.

Pascal

```
function DragEnter(const DataObject: IDataObject; Ke
```

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 Class](#) |
[TVTDragManager.DragEnter](#)
Method |
[TVTDragManager.DragOver](#)
Method

TVTDragManager.DragLeave Method

[TVTDragManager Class](#)

Not documented.

Pascal

```
function DragLeave: HRESULT; stdcall;
```

Description

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

Class

[TVTDragManager Class](#)

Links

[TVTDragManager Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTDragManager Class](#) |
[TVTDragManager.DragLeave
Method](#) | [TVTDragManager.Drop
Method](#)

TVTDragManager.DragOver Method

[TVTDragManager Class](#)

Not documented.

Pascal

```
function DragOver(KeyState: Integer; Pt: TPoint; var
```

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 Class](#) |
[TVTDragManager.DragOver](#)
Method |
[TVTDragManager.ForceDragLeave](#)
Method

TVTDragManager.Drop Method

[TVTDragManager Class](#)

Not documented.

Pascal

```
function Drop(const DataObject: IDataObject; KeyStat
```

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 Class](#) |
[TVTDragManager.Drop Method](#) |
[TVTDragManager.GiveFeedback Method](#)

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 Class](#) |
[TVTDragManager.ForceDragLeave](#)
[Method](#) |
[TVTDragManager.QueryContinueDrag](#)
[Method](#)

TVTDragManager.GiveFeedback Method

[TVTDragManager Class](#)

Not documented.

Pascal

```
function GiveFeedback(Effect: Integer): HRESULT; stdcall
```

Description

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

Class

[TVTDragManager Class](#)

Links

[TVTDragManager Class](#)

What do you think about this topic? [Send feedback!](#)



TVTDragManager.QueryContinueDrag Method

[TVTDragManager Class](#)

Not documented.

Pascal

```
function QueryContinueDrag(EscapePressed: BOOL; KeyS
```

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

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

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.AutoSize Property

TVTEdit Class

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 Class](#) |
[TVTEdit.AutoSelect Property](#) |
[TVTEdit.BorderStyle Property](#)

TVTEdit.AutoSize Property

[TVTEdit Class](#)

Not documented.

Pascal

```
property AutoSize;
```

Description

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

Class

[TVTEdit Class](#)

Links

[TVTEdit Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTEdit Class](#) | [TVTEdit.AutoSize Property](#) | [TVTEdit.CharCase Property](#)

TVTEdit.BorderStyle Property

[TVTEdit Class](#)

Not documented.

Pascal

```
property BorderStyle;
```

Description

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

Class

[TVTEdit Class](#)

Links

[TVTEdit Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTEdit Class](#) |
[TVTEdit.BorderStyle Property](#) |
[TVTEdit.HideSelection Property](#)

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 Class](#) |
[TVTEdit.CharCase Property](#) |
[TVTEdit.MaxLength Property](#)

TVTEdit.HideSelection Property

[TVTEdit Class](#)

Not documented.

Pascal

```
property HideSelection;
```

Description

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

Class

[TVTEdit Class](#)

Links

[TVTEdit Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTEdit Class](#) |
[TVTEdit.HideSelection Property](#) |
[TVTEdit.OEMConvert Property](#)

TVTEdit.MaxLength Property

[TVTEdit Class](#)

Not documented.

Pascal

```
property MaxLength;
```

Description

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

Class

[TVTEdit Class](#)

Links

[TVTEdit Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTEdit Class](#) |
[TVTEdit.MaxLength Property](#) |
[TVTEdit.PasswordChar Property](#)

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 Class](#) |
[TVTEdit.OEMConvert Property](#) |
[TVTEdit.AutoAdjustSize Method](#)

TVTEdit.PasswordChar Property

[TVTEdit Class](#)

Not documented.

Pascal

```
property PasswordChar ;
```

Description

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

Class

[TVTEdit Class](#)

Links

[TVTEdit Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTEdit Class](#) |
[TVTEdit.PasswordChar Property](#) |
[TVTEdit.Create Constructor](#)

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 Class](#) |
[TVTEdit.AutoAdjustSize Method](#) |
[TVTEdit.CreateParams Method](#)

TVTEdit.Create Constructor

[TVTEdit Class](#)

Not documented.

Pascal

```
constructor Create(Link: TStringEditLink); reintrodu
```

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 Class](#) | [TVTEdit.Create
Constructor](#) | [TVTEdit.Release
Method](#)

TVTEdit.CreateParams Method

[TVTEdit Class](#)

Not documented.

Pascal

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

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 position and **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



File

VirtualTrees

Links

[Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TVTHeader.AutoSizeIndex 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 Class](#) |
[TVTHeader.AutoSizeIndex
Property](#) | [TVTHeader.Columns
Property](#)

TVTHeader.Background Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property Background: TColor;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.Background Property](#) |
[TVTHeader.DragImage Property](#)

TVTHeader.Columns Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property Columns: TVirtualTreeColumns;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.Columns Property](#) |
[TVTHeader.Font Property](#)

TVTHeader.DragImage Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property DragImage: TVTDragImage;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.DragImage Property](#) |
[TVTHeader.Height Property](#)

TVTHeader.Font Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property Font: TFont;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.Font Property](#) |
[TVTHeader.Images Property](#)

TVTHeader.Height Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property Height: Cardinal;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.Height Property](#) |
[TVTHeader.MainColumn Property](#)

TVTHeader.Images Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property Images: TCustomImageList;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.Images Property](#) |
[TVTHeader.Options Property](#)

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 Class](#) |
[TVTHeader.MainColumn Property](#)
| [TVTHeader.ParentFont Property](#)

TVTHeader.Options Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property Options: TVTHeaderOptions;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.Options Property](#) |
[TVTHeader.PopupMenu Property](#)

TVTHeader.ParentFont Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property ParentFont: Boolean;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.ParentFont Property](#) |
[TVTHeader.SortColumn Property](#)

TVTHeader.PopupMenu Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property PopupMenu: TPopupMenu;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.PopupMenu Property](#) |
[TVTHeader.SortDirection Property](#)

TVTHeader.SortColumn Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property SortColumn: TColumnIndex;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.SortColumn Property](#) |
[TVTHeader.States Property](#)

TVTHeader.SortDirection Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property SortDirection: TSortDirection;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.SortDirection Property](#)
| [TVTHeader.Style Property](#)

TVTHeader.States Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property States: THeaderStates;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.States Property](#) |
[TVTHeader.Treeview Property](#)

TVTHeader.Style Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property Style: TVTHeaderStyle;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.Style Property](#) |
[TVTHeader.UseColumns Property](#)

TVTHeader.Treeview Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property Treeview: TBaseVirtualTree;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.Treeview Property](#) |
[TVTHeader.Assign Method](#)

TVTHeader.UseColumns Property

[TVTHeader Class](#)

Not documented.

Pascal

```
property UseColumns: Boolean;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.UseColumns Property](#)
| [TVTHeader.AutoFitColumns
Method](#)

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 Class](#) |
[TVTHeader.Assign Method](#) |
[TVTHeader.CanWriteColumns
Method](#)

TVTHeader.AutoFitColumns Method

[TVTHeader Class](#)

Not documented.

Pascal

```
procedure AutoFitColumns(Animated: Boolean = True);
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.AutoFitColumns](#)
[Method](#) |
[TVTHeader.ChangeScale Method](#)

TVTHeader.CanWriteColumns Method

[TVTHeader 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

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.CanWriteColumns](#)
[Method](#) | [TVTHeader.Create](#)
[Constructor](#)

TVTHeader.ChangeScale Method

[TVTHeader Class](#)

Not documented.

Pascal

```
procedure ChangeScale(M: Integer; D: Integer); virtu
```

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 Class](#) |
[TVTHeader.ChangeScale Method](#)
| [TVTHeader.Destroy Destructor](#)

TVTHeader.Create Constructor

[TVTHeader Class](#)

Not documented.

Pascal

```
constructor Create(AOwner: TBaseVirtualTree); virtua
```

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 Class](#) |
[TVTHeader.Create Constructor](#) |
[TVTHeader.DetermineSplitterIndex](#)
Method

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 Class](#) |
[TVTHeader.Destroy Destructor](#) |
[TVTHeader.DragTo Method](#)

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 Class](#) |
[TVTHeader.DetermineSplitterIndex](#)
[Method](#) |
[TVTHeader.GetColumnsClass](#)
[Method](#)

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

```
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 Class](#) |
[TVTHeader.DragTo Method](#) |
[TVTHeader.GetOwner Method](#)

TVTHeader.GetColumnsClass Method

[TVTHeader Class](#)

Returns the class to be used for the actual column implementation.

Pascal

```
function 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 Class](#) |
[TVTHeader.GetColumnsClass](#)
[Method](#) | [TVTHeader.GetShiftState](#)
[Method](#)

TVTHeader.GetOwner Method

[TVTHeader 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

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.GetOwner Method](#) |
[TVTHeader.HandleHeaderMouseMove Method](#)

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 Class](#) |
[TVTHeader.GetShiftState Method](#) |
[TVTHeader.HandleMessage Method](#)

TVTHeader.HandleHeaderMouseMove Method

[TVTHeader Class](#)

Not documented.

Pascal

```
function HandleHeaderMouseMove(var Message: TWMMouse
```

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 Class](#) |
[TVTHeader.HandleHeaderMouseMove](#)
[Method](#) |
[TVTHeader.ImageListChange Method](#)

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 Class](#) |
[TVTHeader.HandleMessage
Method](#) | [TVTHeader.InHeader
Method](#)

TVTHeader.ImageListChange Method

[TVTHeader Class](#)

Not documented.

Pascal

```
procedure ImageListChange(Sender: TObject);
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.ImageListChange
Method](#) | [TVTHeader.Invalidate
Method](#)

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!](#)



[TVTHHeader Class](#) |
[TVTHHeader.InHeader Method](#) |
[TVTHHeader.LoadFromStream Method](#)

TVTHHeader.Invalidate Method

[TVTHHeader Class](#)

Invalidates the entire header or parts of it so they are repainted.

Pascal

```
procedure Invalidate(Column: TVirtualTreeColumn; Exp
```

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

[TVTHHeader Class](#)

Links

[TVTHHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.Invalidate Method](#) |
[TVTHeader.PrepareDrag Method](#)

TVTHeader.LoadFromStream Method

[TVTHeader Class](#)

Restores the state of the header from the given stream.

Pascal

```
procedure LoadFromStream(const Stream: TStream); var
```

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 Class](#) |
[TVTHeader.LoadFromStream](#)
[Method](#) |
[TVTHeader.ReadColumns Method](#)

TVTHeader.PrepareDrag Method

[TVTHeader Class](#)

Initializes dragging of the header, **P** is the current mouse position and **Start** the initial mouse position.

Pascal

```
procedure PrepareDrag(P: TPoint; Start: TPoint);
```

Description

Initializes dragging of the header, **P** is the current mouse position and **Start** the initial mouse position.

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.PrepareDrag Method](#) |
[TVTHeader.RecalculateHeader Method](#)

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 Class](#) |
[TVTHeader.ReadColumns Method](#)
| [TVTHeader.RestoreColumns Method](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.RecalculateHeader](#)
[Method](#) |
[TVTHeader.SaveToStream](#)
[Method](#)

TVTHeader.RestoreColumns Method

[TVTHeader Class](#)

Restores all columns to their width which they had before they have been auto fitted.

Pascal

```
procedure RestoreColumns;
```

Description

Restores all columns to their width which they had before they have been auto fitted.

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeader Class](#) |
[TVTHeader.RestoreColumns](#)
[Method](#) |
[TVTHeader.UpdateMainColumn](#)
[Method](#)

TVTHeader.SaveToStream Method

[TVTHeader Class](#)

Saves the complete state of the header into the provided stream.

Pascal

```
procedure SaveToStream(const Stream: TStream); virtu
```

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 Class](#) |
[TVTHeader.SaveToStream](#)
Method |
[TVTHeader.UpdateSpringColumns](#)
Method

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 Class](#) |
[TVTHeader.UpdateMainColumn](#)
[Method](#) |
[TVTHeader.WriteColumns Method](#)

TVTHeader.UpdateSpringColumns Method

[TVTHeader Class](#)

Not documented.

Pascal

```
procedure UpdateSpringColumns;
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



TVTHeader.WriteColumns Method

TVTHeader Class

Not documented.

Pascal

```
procedure WriteColumns(Writer: TWriter);
```

Description

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

Class

[TVTHeader Class](#)

Links

[TVTHeader Class](#)

What do you think about this topic? [Send feedback!](#)



TVTHeaderPopupMenu Class

[Events](#) | [Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Not documented.

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

 [OnAddHeaderPopupItem](#)

Not documented.

 [OnColumnChange](#)

Not documented.

Properties

 [Options](#)

Not documented.

Methods

 [DoAddHeaderPopupItem](#)

Not documented.

   DoColumnChange

Not documented.

  OnMenuItemClick

Not documented.

   Popup

Not documented.

Legend



published



Event



Property



protected



Method



virtual



public

Class Hierarchy



File

VTHeaderPopup

Links

[Events](#), [Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TVTHeaderPopupMenu Class |
TVTHeaderPopupMenu.OnColumnChange
Event

TVTHeaderPopupMenu.OnAddHeaderPop Event

TVTHeaderPopupMenu Class

Not documented.

Pascal

```
property OnAddHeaderPopupMenu: TAddHeaderPopupMenuEvent;
```

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.OnAddHeaderPopupMenuItem](#)
[Event](#) | [TVTHeaderPopupMenu.Options](#) Property

TVTHeaderPopupMenu.OnColumnChange Event

[TVTHeaderPopupMenu Class](#)

Not documented.

Pascal

```
property OnColumnChange: TColumnChangeEvent;
```

Description

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

Class

[TVTHeaderPopupMenu Class](#)

Links

[TVTHeaderPopupMenu Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeaderPopupMenu Class](#) |
[TVTHeaderPopupMenu.OnColumnChange Event](#) |
[TVTHeaderPopupMenu.DoAddHeaderPopupMenuItem Method](#)

TVTHeaderPopupMenu.Options Property

[TVTHeaderPopupMenu Class](#)

Not documented.

Pascal

```
property Options: TVTHeaderPopupMenuOptions;
```

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.Options Property](#) |
[TVTHeaderPopupMenu.DoColumnChange Method](#)

TVTHeaderPopupMenu.DoAddHeaderPopupMenu

[TVTHeaderPopupMenu Class](#)

Not documented.

Pascal

```
procedure DoAddHeaderPopupMenu(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!](#)



[TVTHaderPopupMenu Class](#) |
[TVTHaderPopupMenu.DoAddHeaderPopupMenuItem](#)
Method |
[TVTHaderPopupMenu.OnMenuItemClick](#)
Method

TVTHaderPopupMenu.DoColumnChange Method

[TVTHaderPopupMenu Class](#)

Not documented.

Pascal

```
procedure DoColumnChange(Column: TColumnIndex; Visible: Boolean);
```

Description

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

Class

[TVTHaderPopupMenu Class](#)

Links

[TVTHaderPopupMenu Class](#)

What do you think about this topic? [Send feedback!](#)



[TVTHeaderPopupMenu Class](#) |
[TVTHeaderPopupMenu.DoColumnChange](#)
[Method](#) | [TVTHeaderPopupMenu.Popup](#)
[Method](#)

TVTHeaderPopupMenu.OnMenuItemClick Method

[TVTHeaderPopupMenu Class](#)

Not documented.

Pascal

```
procedure OnMenuItemClick(Sender: TObject);
```

Description

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

Class

[TVTHeaderPopupMenu Class](#)

Links

[TVTHeaderPopupMenu Class](#)

What do you think about this topic? [Send feedback!](#)



TVTHaderPopupMenu Class |
TVTHaderPopupMenu.OnMenuItemClick
Method

TVTHaderPopupMenu.Popup Method

TVTHaderPopupMenu Class

Not documented.

Pascal

```
procedure Popup(x: Integer; y: Integer); override;
```

Description

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

Class

[TVTHaderPopupMenu Class](#)

Links

[TVTHaderPopupMenu Class](#)

What do you think about this topic? [Send feedback!](#)



[Classes](#) | [TVTHeaderPopupMenu Class](#) | [TWorkerThread Class](#)

TWideBufferedString Class

[Classes](#) | [Methods](#) | [Properties](#) | [Legend](#)

Not documented.

Pascal

```
TWideBufferedString = class;
```

Description

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

Group

[Classes](#)

Members

Properties

   [AsString](#)

Not documented.

Methods

  [Add](#)

Not documented.

  [AddNewLine](#)

Not documented.

   [Destroy](#)

Not documented.

Legend



public



Property



read only



Method



virtual

Class Hierarchy

`TWideBufferedString`

File

VirtualTrees

Links

[Classes](#), [Methods](#), [Properties](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



[TWideBufferedString Class](#) |
[TWideBufferedString.Add Method](#)

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 Class](#) |
[TWideBufferedString.AsString](#)
[Property](#) |
[TWideBufferedString.AddNewLine](#)
[Method](#)

TWideBufferedString.Add Method

[TWideBufferedString Class](#)

Not documented.

Pascal

```
procedure Add(const S: WideString);
```

Description

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

Class

[TWideBufferedString Class](#)

Links

[TWideBufferedString Class](#)

What do you think about this topic? [Send feedback!](#)



[TWideBufferedString Class](#) |
[TWideBufferedString.Add Method](#) |
[TWideBufferedString.Destroy](#)
[Destructor](#)

TWideBufferedString.AddNewLine Method

[TWideBufferedString Class](#)

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 Class](#) |
[TWideBufferedString.AddNewLine](#)
Method

TWideBufferedString.Destroy Destructor

[TWideBufferedString 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

[TWideBufferedString Class](#)

Links

[TWideBufferedString Class](#)

What do you think about this topic? [Send feedback!](#)



[Classes](#) | [TWideBufferedString Class](#) | [TWriterHack Class](#)

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 Class](#) |
[TWorkerThread.AddTree Method](#)

TWorkerThread.CurrentTree Property

[TWorkerThread Class](#)

Not documented.

Pascal

```
property CurrentTree: TBaseVirtualTree;
```

Description

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

Class

[TWorkerThread Class](#)

Links

[TWorkerThread Class](#)

What do you think about this topic? [Send feedback!](#)



[TWorkerThread Class](#) |
[TWorkerThread.CurrentTree](#)
Property |
[TWorkerThread.ChangeTreeStates](#)
Method

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 Class](#) |
[TWorkerThread.AddTree Method](#) |
[TWorkerThread.Create](#)
[Constructor](#)

TWorkerThread.ChangeTreeStates Method

[TWorkerThread Class](#)

Not documented.

Pascal

```
procedure ChangeTreeStates(EnterStates: TChangeState
```

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 Class](#) |
[TWorkerThread.ChangeTreeStates](#)
[Method](#) | [TWorkerThread.Destroy](#)
[Destructor](#)

TWorkerThread.Create Constructor

[TWorkerThread Class](#)

Not documented.

Pascal

```
constructor Create(CreateSuspended: Boolean);
```

Description

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

Class

[TWorkerThread Class](#)

Links

[TWorkerThread Class](#)

What do you think about this topic? [Send feedback!](#)



[TWorkerThread Class](#) |
[TWorkerThread.Create](#)
[Constructor](#) |
[TWorkerThread.Execute Method](#)

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 Class](#) |
[TWorkerThread.Destroy](#)
[Destructor](#) |
[TWorkerThread.RemoveTree](#)
[Method](#)

TWorkerThread.Execute Method

[TWorkerThread Class](#)

Not documented.

Pascal

```
procedure Execute; override;
```

Description

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

Class

[TWorkerThread Class](#)

Links

[TWorkerThread Class](#)

What do you think about this topic? [Send feedback!](#)



[TWorkerThread Class](#) |
[TWorkerThread.Execute Method](#)

TWorkerThread.RemoveTree Method

[TWorkerThread Class](#)

Not documented.

Pascal

```
procedure RemoveTree(Tree: TBaseVirtualTree);
```

Description

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

Class

[TWorkerThread Class](#)

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

Functions

General purpose procedure to blend one bitmap to another.

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; nC
```

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!](#)



Functions | [DrawTextW Function](#) |
[EnumerateVTClipboardFormats](#)
Function

EnumerateVTClipboardFormats Function

[Functions](#)

Not documented.

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!](#)



[Functions](#) |
[EnumerateVTClipboardFormats](#)
[Function](#) |
[GetVTClipboardFormatDescription](#)
[Function](#)

EnumerateVTClipboardFormats Function

[Functions](#)

Not documented.

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

Functions

Not documented.

Pascal

```
function GetVTClipboardFormatDescription(AFormat: Wo
```

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!](#)



[Functions](#) |
[GetVTClipboardFormatDescription](#)
[Function](#) |
[RegisterVTClipboardFormat](#)
[Function](#)

PrtStretchDrawDIB Function

[Functions](#)

Not documented.

Pascal

```
procedure PrtStretchDrawDIB(Canvas: TCanvas; DestRec
```

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!](#)



[Functions](#) | [PrtStretchDrawDIB Function](#) | [ShortenString Function](#)

RegisterVTClipboardFormat Function

[Functions](#)

Methods to register a certain clipboard format for a given tree class.

Pascal

```
procedure RegisterVTClipboardFormat(AFormat: Word; T  
function RegisterVTClipboardFormat(Description: stri
```

Description

Registration with the clipboard is done here too and the assigned ID returned by the function. tyled 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

Functions

General purpose routine to shorten a Unicode string to a given maximum size.

Pascal

```
function ShortenString(DC: HDC; const S: 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 explicitly. **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

File

VirtualTrees

Links

[Functions](#)

What do you think about this topic? [Send feedback!](#)



TreeFromNode Function

Functions

General purpose routine to get the tree to which a node belongs.

Pascal

```
function TreeFromNode(Node: PVirtualNode): TBaseVirt
```

Description

For obvious reasons it makes no sense to store the reference to a tree in each node record, but sometimes there might arise the need to know to which tree a node belongs. This is not often the case but is necessary e.g. for optimized moves in drag'n drop or cut'n paste operations.

Each node contains a reference to its parent to allow fast traversal. The hidden root node, however, does not need this reference because it does not have a node parent. Instead it contains the reference of the tree to which it belongs. To determine which node is the root node (when you don't know its tree) a special case of sibling reference is used. Since the root node does neither have a previous nor a next sibling the corresponding pointers are set to the root node, making the root so pointing to itself. This case will never happen in "normal" nodes, so it is a reliable way to detect the root node.

Group

Functions

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.

 [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



Struct

Links

[Enumerations](#), [Virtual Treeview](#), [Records](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



TBaseChunk Record

Structs and Records

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

Structs and Records

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!](#)



[Structs and Records](#) |
[TBaseChunkBody Record](#) |
[TChunkHeader Record](#)

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!](#)



[Structs and Records](#) |
[TCacheEntry Record](#) |
[TClipboardFormatEntry Record](#)

TChunkHeader Record

[Structs and Records](#)

Not documented.

Pascal

```
TChunkHeader = record  
  ChunkSize: Integer;  
  ChunkType: Integer;  
end;
```

Description

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

Group

[Structs and Records](#)

File

VirtualTrees

Links

[Structs and Records](#)

What do you think about this topic? [Send feedback!](#)



Structs and Records |
TChunkHeader Record |
TClipboardFormatListEntry Record

TClipboardFormatEntry Record

Structs and Records

Not documented.

Pascal

```
TClipboardFormatEntry = record  
  ID: Word;  
  Description: string;  
end;
```

Description

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

Group

Structs and Records

File

VirtualTrees

Links

Structs and Records

What do you think about this topic? [Send feedback!](#)



TClipboardFormatListEntry Record

Structs and Records

Not documented.

Pascal

```
TClipboardFormatListEntry = record  
  Description: string;  
  TreeClass: TVirtualTreeClass;  
  Priority: Cardinal;  
  FormatEtc: TFormatEtc;  
end;
```

Description

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

Members

Members	Description
Description: string;	The string used to register the format with Windows.
TreeClass: TVirtualTreeClass;	The tree class which supports rendering this format.
Priority: Cardinal;	Number which determines the order of formats used in IDataObject.
FormatEtc: TFormatEtc;	The definition of the format in the IDataObject.

Group

Structs and Records

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!](#)



[Structs and Records](#) |
[THeaderPaintInfo Record](#) |
[TInternalStgMedium Record](#)

THitInfo Record

[Structs and Records](#)

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

Structs and Records

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!](#)



[Structs and Records](#) |
[TInternalStgMedium Record](#) |
[TSHDragImage Record](#)

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!](#)



[Structs and Records](#) |
[TRealWMNCPaint Record](#) |
[TToggleAnimationData Record](#)

TSHDragImage Record

[Structs and Records](#)

Not documented.

Pascal

```
TSHDragImage = packed record
  sizeDragImage: TSize;
  ptOffset: TPoint;
  hbmpDragImage: HBITMAP;
  ColorRef: TColorRef;
end;
```

Description

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

Group

[Structs and Records](#)

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

Members	Description
Expand: Boolean;	if true then expanding is in progress
Window: HWND;	copy of the tree's window handle
DC: HDC;	the DC of the window to erase uncovered parts
Brush: HBRUSH;	the brush to be used to erase uncovered parts
R: TRect;	the scroll rectangle

Group

Structs and Records

File

VirtualTrees

Links

Structs and Records

What do you think about this topic? [Send feedback!](#)



TVirtualNode Record

Structs and Records

Not documented.

Pascal

```
TVirtualNode = packed record  
  ChildCount: Cardinal;  
  Index: Cardinal;  
  NodeHeight: Word;  
  States: TVirtualNodeStates;  
  Align: Byte;  
  CheckState: TCheckState;  
  CheckType: TCheckType;  
  Dummy: Byte;  
  TotalCount: Cardinal;  
  TotalHeight: Cardinal;  
  FirstChild: PVirtualNode;  
  LastChild: PVirtualNode;  
  NextSibling: PVirtualNode;  
  Parent: PVirtualNode;  
  PrevSibling: PVirtualNode;  
  Data: record;  
  end;
```

Description

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

Members

Members	Description
---------	-------------

NodeHeight: Word;	height in pixels
States: TVirtualNodeStates;	states describing various properties of the node (expanded, initialized etc.)
Align: Byte;	line/button alignment
CheckState: TCheckState;	checked, pressed etc.)
CheckType: TCheckType;	indicates which check type shall be used for this node
Dummy: Byte;	dummy value to fill DWORD boundary
Data: record;	this is a placeholder, each node gets extra data determined by NodeDataSize

Group

[Structs and Records](#)

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

Members	Description
HintRect: TRect;	used for draw trees only, string trees get the size from the hint string
DefaultHint: WideString;	used only if there is no node specific hint string available or a header hint is about to appear

```
HintText: WideString;
```

set when size of the hint window
is calculated

Group

Structs and Records

File

VirtualTrees

Links

Structs and Records

What do you think about this topic? [Send feedback!](#)



TVTImageInfo Record

Structs and Records

Not documented.

Pascal

```
TVTImageInfo = record  
  Index: Integer;  
  XPos: Integer;  
  YPos: Integer;  
  Ghosted: Boolean;  
  Images: TCustomImageList;  
end;
```

Description

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

Members

Members	Description
Index: Integer;	index in the associated image list
Ghosted: Boolean;	flag to indicate that the image must be drawn slightly lighter

Group

Structs and Records

File

VirtualTrees

Links

Structs and Records

What do you think about this topic? [Send feedback!](#)



TVTPaintInfo Record

Structs and Records

Not documented.

Pascal

```
TVTPaintInfo = record
  Canvas: TCanvas;
  PaintOptions: TVTInternalPaintOptions;
  Node: PVirtualNode;
  Column: TColumnIndex;
  Position: TColumnPosition;
  CellRect: TRect;
  ContentRect: TRect;
  NodeWidth: Integer;
  Alignment: TAlignment;
  BidiMode: TBidiMode;
  BrushOrigin: TPoint;
  ImageInfo: array[TVTImageInfoIndex] of TVTImageInfo;
end;
```

Description

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

Members

Members	Description
Canvas: TCanvas;	the canvas to paint on
PaintOptions: TVTInternalPaintOptions;	a copy of the paint options passed to PaintTree

Node: PVirtualNode;	the node to paint
Column: TColumnIndex;	the node's column index to paint
Position: TColumnPosition;	the column position of the node
Nodewidth: Integer;	the actual node width
Alignment: TAlignment;	how to align within the node rectangle
BidiMode: TBidiMode;	directionality to be used for painting
BrushOrigin: TPoint;	the alignment for the brush used to draw dotted lines
ImageInfo: array[TVTImageInfoIndex] of TVTImageInfo;	info about each possible node image

Group

[Structs and Records](#)

File

VirtualTrees

Links

[Structs and Records](#)

What do you think about this topic? [Send feedback!](#)



[Structs and Records](#) |
[TVTPaintInfo Record](#) |
[TVTTooltipLineBreakStyle](#)
[Enumeration](#)

TVTReference Record

[Structs and Records](#)

Not documented.

Pascal

```
TVTReference = record  
  Process: Cardinal;  
  Tree: TBaseVirtualTree;  
end;
```

Description

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

Group

[Structs and Records](#)

File

VirtualTrees

Links

[Structs and Records](#)

What do you think about this topic? [Send feedback!](#)



TVTTooltipLineBreakStyle Enumeration

Structs and Records

Not documented.

Pascal

```
TVTTooltipLineBreakStyle = (h1bDefault, h1bForceSing
```

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!](#)



Types

[Enumerations](#) | [Virtual Treeview](#) | [Types](#) | [Legend](#)

These are all types that are contained in this documentation.

Enumerations

❖ TAddPopuItem

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.

Group

Virtual Treeview

Types

- ❖ **PCardinal**
Not documented.
- ❖ **PClipboardFormatListEntry**
Not documented.
- ❖ **PSHDragImage**
Not documented.
- ❖ **PVirtualNode**
Not documented.
- ❖ **PVTHintData**
Not documented.
- ❖ **PVTReference**
Not documented.
- ❖ **TAddHeaderPopuItemEvent**
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.

◆ TVTColumnDbClickEvent

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.

 **TVTGetCellsIsEmptyEvent**

Not documented.

 **TVTGetImageExEvent**

Not documented.

 **TVTMenuItem**

Not documented.

 **TVTScrollbarShowEvent**

Not documented.

Legend



Type
 Struct

Links

[Enumerations](#), [Virtual Treeview](#), [Types](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



PCardinal Type

Types

Not documented.

Pascal

```
PCardinal = ^Cardinal;
```

Description

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

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | PCardinal Type |
PSHDragImage Type

PClipboardFormatListEntry Type

Types

Not documented.

Pascal

```
PClipboardFormatListEntry = ^TClipboardFormatListEnt
```

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!](#)



[Types](#) | [PClipboardFormatListEntry Type](#) | [PVirtualNode Type](#)

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!](#)



[Types](#) | [PSHDragImage Type](#) | [PVTHintData Type](#)

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!](#)



[Types](#) | [PVirtualNode Type](#) | [PVTReference Type](#)

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!](#)



[Types](#) | [PVTHintData Type](#) | [TAddHeaderPopuItemEvent Type](#)

PVTReference Type

[Types](#)

Not documented.

Pascal

```
PVTReference = ^TVTReference;
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



[Types](#) | [PVTReference Type](#) |
[TAutoScrollInterval Type](#)

TAddHeaderPopupItemEvent Type

[Types](#)

Not documented.

Pascal

```
TAddHeaderPopupItemEvent = procedure (const Sender :
```

Description

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

Group

[Types](#)

File

VTHaderPopup

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!](#)



[Types](#) | [TAutoScrollInterval Type](#) | [TCardinalArray Type](#)

TCache Type

[Types](#)

Not documented.

Pascal

```
TCache = array of TCacheEntry;
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TCache Type](#) |
[TChangeStates Type](#)

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!](#)



[Types](#) | [TCardinalArray Type](#) | [TColumnChangeEvent Type](#)

TChangeStates Type

[Types](#)

Not documented.

Pascal

```
TChangeStates = set of ( csStopValidation, csUseCach
```

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!](#)



[Types](#) | [TChangeStates Type](#) |
[TColumnIndex Type](#)

TColumnChangeEvent Type

[Types](#)

Not documented.

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!](#)



Types | TColumnChangeEvent
Type | TColumnPosition Type

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!](#)



[Types](#) | [TColumnIndex Type](#) | [TColumnsArray Type](#)

TColumnPosition Type

[Types](#)

Not documented.

Pascal

```
TColumnPosition = type Cardinal;
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TColumnPosition Type](#) |
[TDragOperations Type](#)

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!](#)



[Types](#) | [TColumnsArray Type](#) | [TFormatArray Type](#)

TDragOperations Type

[Types](#)

Not documented.

Pascal

```
TDragOperations = set of TDragOperation;
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TDragOperations Type](#) | [TFormatEtcArray Type](#)

TFormatArray Type

[Types](#)

Not documented.

Pascal

```
TFormatArray = array of Word;
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TFormatArray Type](#) | [TGetFirstNodeProc Type](#)

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!](#)



[Types](#) | [TFormatEtcArray Type](#) |
[TGetNextNodeProc Type](#)

TGetFirstNodeProc Type

[Types](#)

Not documented.

Pascal

```
TGetFirstNodeProc = function : PVirtualNode of objec
```

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!](#)



[Types](#) | [TGetFirstNodeProc Type](#) | [THeaderPaintElements Type](#)

TGetNextNodeProc Type

[Types](#)

Not documented.

Pascal

```
TGetNextNodeProc = function (Node: PVirtualNode): PV
```

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!](#)



[Types](#) | [TGetNextNodeProc Type](#) | [THeaderStates Type](#)

THeaderPaintElements Type

[Types](#)

Not documented.

Pascal

```
THeaderPaintElements = set of ( hpeBackground, hpeDr
```

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!](#)



[Types](#) | [THeaderPaintElements
Type](#) | [THitPositions Type](#)

THeaderStates Type

[Types](#)

Not documented.

Pascal

```
THeaderStates = set of THeaderState;
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



[Types](#) | [THeaderStates Type](#) |
[TImageIndex Type](#)

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!](#)



[Types](#) | [THitPositions Type](#) |
[TIndexArray Type](#)

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!](#)



[Types](#) | [TImageIndex Type](#) | [TInternalStgMediumArray Type](#)

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

```
TInternalStgMediumArray = array of TInternalStgMediu
```

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!](#)



[Types](#) | [TInternalStgMediumArray
Type](#) | [TMagicID Type](#)

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!](#)



[Types](#) | [TLineImage Type](#) | [TMouseButton Type](#)

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!](#)



[Types](#) | [TMagicID Type](#) | [TNodeArray Type](#)

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!](#)



[Types](#) | [TMouseButton Type](#) | [TScrollDirections Type](#)

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!](#)



[Types](#) | [TNodeArray Type](#) | [TScrollUpdateOptions Type](#)

TScrollDirections Type

[Types](#)

Not documented.

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!](#)



[Types](#) | [TScrollDirections Type](#) | [TTreeOptionsClass Type](#)

TScrollUpdateOptions Type

[Types](#)

Not documented.

Pascal

```
TScrollUpdateOptions = set of ( suoRepaintHeader, su
```

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!](#)



[Types](#) | [TScrollUpdateOptions](#)
[Type](#) | [TVirtualNodeInitStates](#) [Type](#)

TTreeOptionsClass Type

[Types](#)

Not documented.

Pascal

```
TTreeOptionsClass = class of TCustomVirtualTreeOptio
```

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!](#)



Types | TTreeOptionsClass Type |
TVirtualNodeStates Type

TVirtualNodeInitStates Type

Types

Not documented.

Pascal

```
TVirtualNodeInitStates = set of TVirtualNodeInitStat
```

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!](#)



Types | TVirtualNodeInitStates
Type | TVirtualTreeClass Type

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!](#)



[Types](#) | [TVirtualNodeStates Type](#) |
[TVirtualTreeColumnClass Type](#)

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!](#)



[Types](#) | [TVirtualTreeClass Type](#) |
[TVirtualTreeColumnsClass Type](#)

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!](#)



Types | TVirtualTreeColumnClass
Type | TVirtualTreeStates Type

TVirtualTreeColumnsClass Type

Types

Not documented.

Pascal

```
TVirtualTreeColumnsClass = class of TVirtualTreeColu
```

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!](#)



[Types](#) | [TVirtualTreeColumnsClass](#)
[Type](#) | [TVSTGetTextEvent](#) [Type](#)

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!](#)



[Types](#) | [TVirtualTreeStates Type](#) |
[TVSTNewTextEvent Type](#)

TVSTGetTextEvent Type

[Types](#)

Not documented.

Pascal

```
TVSTGetTextEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



Types | [TVSTGetTextEvent Type](#) |
[TVSTShortenStringEvent Type](#)

TVSTNewTextEvent Type

Types

Not documented.

Pascal

```
TVSTNewTextEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



Types | [TVSTNewTextEvent Type](#) |
[TVTAdvancedHeaderPaintEvent](#)
[Type](#)

TVSTShortenStringEvent Type

Types

Not documented.

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!](#)



Types | TVSTShortenStringEvent
Type | TVTAfterCellPaintEvent
Type

TVTAdvancedHeaderPaintEvent Type

Types

Not documented.

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!](#)



Types |
TVTAdvancedHeaderPaintEvent
Type | TVTAfterItemEraseEvent
Type

TVTAfterCellPaintEvent Type

Types

Not documented.

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!](#)



Types | TVTAfterCellPaintEvent
Type | TVTAfterItemPaintEvent
Type

TVTAfterItemEraseEvent Type

Types

Not documented.

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!](#)



Types | TVTAfterItemEraseEvent
Type | TVTAnimationCallback
Type

TVTAfterItemPaintEvent Type

Types

Not documented.

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!](#)



[Types](#) | [TVTAfterItemPaintEvent](#)
[Type](#) | [TVTAnimationOptions](#) [Type](#)

TVTAnimationCallback Type

[Types](#)

Not documented.

Pascal

```
TVTAnimationCallback = function (Step, StepSize: Int
```

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!](#)



[Types](#) | [TVTAnimationCallback
Type](#) | [TVTAutoOptions Type](#)

TVTAnimationOptions Type

[Types](#)

Not documented.

Pascal

```
TVTAnimationOptions = set of TVTAnimationOption;
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



Types | TVTAnimationOptions
Type | TVTBackgroundPaintEvent
Type

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!](#)



[Types](#) | [TVTAutoOptions Type](#) | [TVTBeforeCellPaintEvent Type](#)

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!](#)



[Types](#) |
[TVTBackgroundPaintEvent Type](#) |
[TVTBeforeItemEraseEvent Type](#)

TVTBeforeCellPaintEvent Type

[Types](#)

Not documented.

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!](#)



Types | TVTBeforeCellPaintEvent
Type | TVTBeforeItemPaintEvent
Type

TVTBeforeItemEraseEvent Type

Types

Not documented.

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!](#)



[Types](#) | [TVTBeforeItemEraseEvent
Type](#) | [TVTBias Type](#)

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!](#)



Types | [TVTBeforeItemPaintEvent
Type](#) | [TVTChangeEvent
Type](#)

TVTBias Type

Types

Not documented.

Pascal

```
TVTBias = -128..127;
```

Description

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

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TVTBias Type](#) | [TVTChangingEvent Type](#)

TVTChangeEvent Type

[Types](#)

Not documented.

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!](#)



[Types](#) | [TVTChangeEvent Type](#) |
[TVTCheckChangingEvent Type](#)

TVTChangingEvent Type

[Types](#)

Not documented.

Pascal

```
TVTChangingEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



Types | [TVTChangingEvent Type](#) |
[TVTColumnClickEvent Type](#)

TVTCheckChangingEvent Type

Types

Not documented.

Pascal

```
TVTCheckChangingEvent = 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!](#)



Types | [TVTCheckChangingEvent](#)
Type | [TVTColumnDbClickEvent](#)
Type

TVTColumnClickEvent Type

Types

Not documented.

Pascal

```
TVTColumnClickEvent = procedure (Sender: TBaseVirtua
```

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!](#)



[Types](#) | [TVTColumnClickEvent Type](#) | [TVTColumnOptions Type](#)

TVTColumnDbClickEvent Type

[Types](#)

Not documented.

Pascal

```
TVTColumnDbClickEvent = 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!](#)



Types | [TVTColumnDbClickEvent](#)
Type | [TVTCompareEvent](#) Type

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!](#)



[Types](#) | [TVTColumnOptions Type](#) |
[TVTCreateDataObjectEvent Type](#)

TVTCompareEvent Type

[Types](#)

Not documented.

Pascal

```
TVTCompareEvent = procedure (Sender: TBaseVirtualTre
```

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!](#)



Types | [TVTCompareEvent Type](#) |
[TVTCreateDragManagerEvent](#)
Type

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!](#)



[Types](#) |
[TVTCreateDataObjectEvent Type](#) |
[TVTCreateEditorEvent Type](#)

TVTCreateDragManagerEvent Type

[Types](#)

Not documented.

Pascal

```
TVTCreateDragManagerEvent = 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!](#)



[Types](#) |
[TVTCreateDragManagerEvent](#)
[Type](#) | [TVTDragAllowedEvent](#)
[Type](#)

TVTCreateEditorEvent Type

[Types](#)

Not documented.

Pascal

```
TVTCreateEditorEvent = procedure (Sender: TBaseVirtu
```

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!](#)



[Types](#) | [TVTCreateEditorEvent Type](#) | [TVTDragDropEvent Type](#)

TVTDragAllowedEvent Type

[Types](#)

Not documented.

Pascal

```
TVTDragAllowedEvent = procedure (Sender: TBaseVirtua
```

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!](#)



[Types](#) | [TVTDragAllowedEvent Type](#) | [TVTDragImageStates Type](#)

TVTDragDropEvent Type

[Types](#)

Not documented.

Pascal

```
TVTDragDropEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



[Types](#) | [TVTDragDropEvent Type](#) | [TVTDragOverEvent Type](#)

TVTDragImageStates Type

[Types](#)

Not documented.

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!](#)



[Types](#) | [TVTDragImageStates](#)
[Type](#) | [TVTDrawHintEvent](#) [Type](#)

TVTDragOverEvent Type

[Types](#)

Not documented.

Pascal

```
TVTDragOverEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



[Types](#) | [TVTDragOverEvent Type](#) |
[TVTDrawNodeEvent Type](#)

TVTDrawHintEvent Type

[Types](#)

Not documented.

Pascal

```
TVTDrawHintEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



Types | [TVTDrawHintEvent Type](#) |
[TVTEditCancelEvent Type](#)

TVTDrawNodeEvent Type

Types

Not documented.

Pascal

```
TVTDrawNodeEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



Types | [TVTDrawNodeEvent Type](#)
| [TVTEditChangeEvent Type](#)

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!](#)



Types | [TVTEditCancelEvent Type](#)
| [TVTEditChangingEvent Type](#)

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!](#)



Types | TVTEditChangeEvent
Type | TVTFocusChangeEvent
Type

TVTEditChangingEvent Type

Types

Not documented.

Pascal

```
TVTEditChangingEvent = procedure (Sender: TBaseVirtu
```

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!](#)



Types | TVTEditChangingEvent
Type | TVTFocusChangingEvent
Type

TVTFocusChangeEvent Type

Types

Not documented.

Pascal

```
TVTFocusChangeEvent = procedure (Sender: TBaseVirtua
```

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!](#)



Types | [TVTFocusChangeEvent](#)
Type | [TVTFreeNodeEvent](#) Type

TVTFocusChangingEvent Type

Types

Not documented.

Pascal

```
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!](#)



Types | [TVTFocusChangingEvent](#)
Type | [TVTGetCursorEvent](#) Type

TVTFreeNodeEvent Type

Types

Not documented.

Pascal

```
TVTFreeNodeEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



[Types](#) | [TVTFreeNodeEvent Type](#) |
[TVTGetHeaderCursorEvent Type](#)

TVTGetCursorEvent Type

[Types](#)

Not documented.

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!](#)



Types | [TVTGetCursorEvent Type](#)
| [TVTGetHintSizeEvent Type](#)

TVTGetHeaderCursorEvent Type

Types

Not documented.

Pascal

```
TVTGetHeaderCursorEvent = procedure (Sender: TVTHead
```

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!](#)



[Types](#) |
[TVTGetHeaderCursorEvent Type](#) |
[TVTGetImageEvent Type](#)

TVTGetHintSizeEvent Type

[Types](#)

Not documented.

Pascal

```
TVTGetHintSizeEvent = procedure (Sender: TBaseVirtua
```

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!](#)



Types | [TVTGetHintSizeEvent](#)
Type | [TVTGetLineStyleEvent](#)
Type

TVTGetImageEvent Type

Types

Not documented.

Pascal

```
TVTGetImageEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



[Types](#) | [TVTGetImageEvent Type](#) |
[TVTGetNodeDataSizeEvent Type](#)

TVTGetLineStyleEvent Type

[Types](#)

Not documented.

Pascal

```
TVTGetLineStyleEvent = procedure (Sender: TBaseVirtu
```

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!](#)



Types | [TVTGetLineStyleEvent](#)
Type | [TVTGetNodeProc](#) Type

TVTGetNodeDataSizeEvent Type

Types

Not documented.

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!](#)



[Types](#) |
[TVTGetNodeDataSizeEvent Type](#) |
[TVTGetNodeWidthEvent Type](#)

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!](#)



[Types](#) | [TVTGetNodeProc Type](#) |
[TVTGetUserClipboardFormatsEvent](#)
[Type](#)

TVTGetNodeWidthEvent Type

[Types](#)

Not documented.

Pascal

```
TVTGetNodeWidthEvent = procedure (Sender: TBaseVirtu
```

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!](#)



Types | [TVTGetNodeWidthEvent](#)
Type | [TVTHeaderClass](#) Type

TVTGetUserClipboardFormatsEvent Type

Types

Not documented.

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!](#)



[Types](#) |
[TVTGetUserClipboardFormatsEvent](#)
[Type](#) | [TVTHeaderClickEvent](#) [Type](#)

TVTHeaderClass Type

[Types](#)

Not documented.

Pascal

```
TVTHeaderClass = class of TVTHeader;
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TVTHeaderClass Type](#) | [TVTHeaderDraggedEvent Type](#)

TVTHeaderClickEvent Type

[Types](#)

Not documented.

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!](#)



Types | [TVTHHeaderClickEvent](#)
Type |
[TVTHHeaderDraggedOutEvent](#)
Type

TVTHHeaderDraggedEvent Type

Types

Not documented.

Pascal

```
TVTHHeaderDraggedEvent = procedure (Sender: 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!](#)



Types | TVTHeaderDraggedEvent
Type | TVSTGetHintEvent Type

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!](#)



[Types](#) |
[TVTHdrDraggedOutEvent](#)
[Type](#) | [TVTHdrDraggingEvent](#)
[Type](#)

TVSTGetHintEvent Type

[Types](#)

Not documented.

Pascal

```
TVSTGetHintEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



[Types](#) | [TVSTGetHintEvent Type](#) |
[TVTHHeaderMouseEvent Type](#)

TVTHHeaderDraggingEvent Type

[Types](#)

Not documented.

Pascal

```
TVTHHeaderDraggingEvent = procedure (Sender: 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!](#)



Types | TVTHeaderDraggingEvent
Type |
TVTHeaderMouseMoveEvent
Type

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!](#)



Types | TVTHeaderMouseEvent
Type | TVTHeaderNotifyEvent
Type

TVTHeaderMouseMoveEvent Type

Types

Not documented.

Pascal

```
TVTHeaderMouseMoveEvent = procedure (Sender: TVTHead
```

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!](#)



[Types](#) |
[TVTHeaderMouseMoveEvent](#)
[Type](#) | [TVTHeaderOptions](#) [Type](#)

TVTHeaderNotifyEvent Type

[Types](#)

Not documented.

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!](#)



Types | TVTHeaderNotifyEvent
Type | TVTHeaderPaintEvent Type

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!](#)



[Types](#) | [TVTHeaderOptions Type](#) |
[TVTHeaderPaintQueryElementsEvent](#)
[Type](#)

TVTHeaderPaintEvent Type

[Types](#)

Not documented.

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!](#)



Types | TVTHeaderPaintEvent
Type | TVTHeaderPopupOptions
Type

TVTHeaderPaintQueryElementsEvent Type

Types

Not documented.

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!](#)



Types |
TVTHelpPaintQueryElementsEvent
Type | TVTHelpContextEvent Type

TVTHelpPopupOptions Type

Types

Not documented.

Pascal

```
TVTHelpPopupOptions = set of TVTHelpPopupOption;
```

Description

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

Group

Types

File

TVTHelpPopup

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTHeaderPopupOptions
Type | TVTHotNodeChangeEvent
Type

TVTHelpContextEvent Type

Types

Not documented.

Pascal

```
TVTHelpContextEvent = procedure (Sender: TBaseVirtua
```

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!](#)



Types | [TVTHelpContextEvent](#)
Type |
[TVTIncrementalSearchEvent](#) Type

TVTHotNodeChangeEvent Type

Types

Not documented.

Pascal

```
TVTHotNodeChangeEvent = 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!](#)



Types | [TVTHotNodeChangeEvent](#)
Type | [TVTInitChildrenEvent](#) Type

TVTIncrementalSearchEvent Type

Types

Not documented.

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!](#)



[Types](#) |
[TVTIncrementalSearchEvent Type](#)
| [TVTInitNodeEvent Type](#)

TVTInitChildrenEvent Type

[Types](#)

Not documented.

Pascal

```
TVTInitChildrenEvent = procedure (Sender: TBaseVirtu
```

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!](#)



Types | [TVTInitChildrenEvent Type](#)
| [TVTInternalPaintOptions Type](#)

TVTInitNodeEvent Type

Types

Not documented.

Pascal

```
TVTInitNodeEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



[Types](#) | [TVTInitNodeEvent Type](#) |
[TVTKeyActionEvent Type](#)

TVTInternalPaintOptions Type

[Types](#)

Not documented.

Pascal

```
TVTInternalPaintOptions = set of TVTInternalPaintOpt
```

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!](#)



Types | [TVTInternalPaintOptions](#)
Type | [TVTMeasureItemEvent](#)
Type

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!](#)



[Types](#) | [TVTKeyActionEvent Type](#) |
[TVTMiscOptions Type](#)

TVTMeasureItemEvent Type

[Types](#)

Not documented.

Pascal

```
TVTMeasureItemEvent = procedure (Sender: TBaseVirtua
```

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!](#)



Types | TVTMeasureItemEvent
Type | TVTNodeCopiedEvent Type

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!](#)



[Types](#) | [TVTMiscOptions Type](#) |
[TVTNodeCopyingEvent Type](#)

TVTNodeCopiedEvent Type

[Types](#)

Not documented.

Pascal

```
TVTNodeCopiedEvent = 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!](#)



[Types](#) | [TVTNodeCopiedEvent Type](#) | [TVTNodeMovedEvent Type](#)

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!](#)



[Types](#) | [TVTNodeCopyingEvent](#)
[Type](#) | [TVTNodeMovingEvent](#) [Type](#)

TVTNodeMovedEvent Type

[Types](#)

Not documented.

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!](#)



Types | [TVTNodeMovedEvent](#)
Type | [TVTPaintEvent](#) Type

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!](#)



Types | [TVTNodeMovingEvent](#)
Type | [TVTPaintOptions](#) Type

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!](#)



[Types](#) | [TVTPaintEvent Type](#) |
[TVTPaintText Type](#)

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!](#)



[Types](#) | [TVTPaintOptions Type](#) |
[TVTPopupEvent Type](#)

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!](#)



[Types](#) | [TVTPaintText Type](#) | [TVTRenderOLEDataEvent Type](#)

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!](#)



[Types](#) | [TVTPopupEvent Type](#) |
[TVTSaveNodeEvent Type](#)

TVTRenderOLEDataEvent Type

[Types](#)

Not documented.

Pascal

```
TVTRenderOLEDataEvent = 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!](#)



[Types](#) | [TVTRenderOLEDataEvent Type](#) | [TVTScrollEvent Type](#)

TVTSaveNodeEvent Type

[Types](#)

Not documented.

Pascal

```
TVTSaveNodeEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



Types | [TVTSaveNodeEvent Type](#)
| [TVTScrollIncrement Type](#)

TVTScrollEvent Type

Types

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!](#)



Types | [TVTScrollEvent Type](#) |
[TVTSelectionOptions Type](#)

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!](#)



[Types](#) | [TVTScrollIncrement Type](#) |
[TVTStateChangeEvent Type](#)

TVTSelectionOptions Type

[Types](#)

Not documented.

Pascal

```
TVTSelectionOptions = set of TVTSelectionOption;
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TVTSelectionOptions Type](#)
| [TVTStringOptions Type](#)

TVTStateChangeEvent Type

[Types](#)

Not documented.

Pascal

```
TVTStateChangeEvent = procedure (Sender: TBaseVirtua
```

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!](#)



Types | TVTStateChangeEvent
Type | TVTStructureChangeEvent
Type

TVTStringOptions Type

Types

Not documented.

Pascal

```
TVTStringOptions = set of TVTStringOption;
```

Description

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

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TVTStringOptions Type](#) |
[TVTTransparency Type](#)

TVTStructureChangeEvent Type

[Types](#)

Not documented.

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!](#)



Types | TVTStructureChangeEvent
Type | TVTUpdatingEvent Type

TVTTransparency Type

Types

Not documented.

Pascal

```
TVTTransparency = 0..255;
```

Description

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

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TVTTransparency Type](#) |
[TWMContextMenu Type](#)

TVTUpdatingEvent Type

[Types](#)

Not documented.

Pascal

```
TVTUpdatingEvent = procedure (Sender: TBaseVirtualTr
```

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!](#)



[Types](#) | [TVTUpdatingEvent Type](#) | [TWMPrintClient Type](#)

TWMContextMenu Type

[Types](#)

Not documented.

Pascal

```
TWMContextMenu = TWMouse;
```

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!](#)



[Types](#) | [TWMContextMenu Type](#) | [TAddPopuItem Type Enumeration](#)

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!](#)



Types | TWMPrintClient Type |
TBlendMode Enumeration

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!](#)



Types | TAddPopuItem Type
Enumeration |
TVTGetCellsEmptyEvent Type

TBlendMode Enumeration

Types

Not documented.

Pascal

```
TBlendMode = (bmConstantAlpha, bmPerPixelAlpha, bmMa
```

Description

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

Members

Members	Description
bmConstantAlpha	apply given constant alpha
bmPerPixelAlpha	use alpha value of the source pixel
bmMasterAlpha	use alpha value of source pixel and multiply it with the constant alpha value
bmConstantAlphaAndColor	blend the destination color with the given constant color und the constant alpha value

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TBlendMode Enumeration
| TChangeReason Enumeration

TVTGetCellsEmptyEvent 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!](#)



Types | TVTGetCellsEmptyEvent
Type | TCheckImageKind
Enumeration

TChangeReason Enumeration

Types

Not documented.

Pascal

```
TChangeReason = (crIgnore, crAccumulated, crChildAdd
```

Description

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

Members

Members	Description
crIgnore	used as placeholder
crAccumulated	used for delayed changes
crChildAdded	one or more child nodes have been added
crChildDeleted	one or more child nodes have been deleted
crNodeAdded	a node has been added
crNodeCopied	a node has been duplicated
crNodeMoved	a node has been moved to a new place

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TCheckImageKind Enumeration

Types

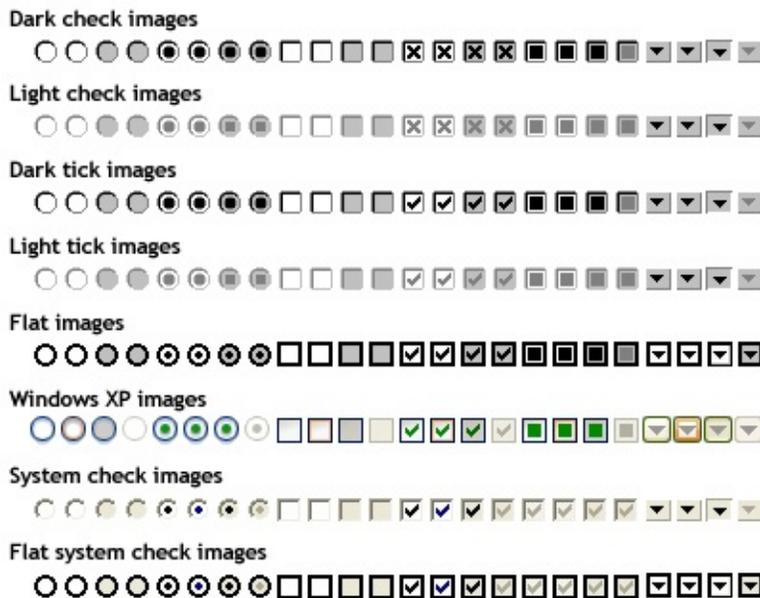
Determines which images should be used for checkboxes and radio buttons.

Pascal

```
TCheckImageKind = (ckLightCheck, ckDarkCheck, ckLigh
```

Description

Provided with the tree are nine different image sets for the check images used when toCheckSupport is enabled in TreeOptions.



Eight of the nine lists are predefined while one is a custom

check image list, which can be filled by the application. Use `ckCustom` as `CheckImageKind` value and assign an image list to the `CustomCheckImages` property to enable custom images.

The order of the images in the image lists is always as listed below. Make sure you have the same amount of images in your custom image list, if you want own check images.

- empty image (`ckEmpty`)

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

Members	Description
ckLightCheck	gray cross
ckDarkCheck	black cross
ckLightTick	gray tick mark
ckDarkTick	black tick mark
ckFlat	flat images (no 3D border)
ckXP	Windows XP style
ckCustom	application defined check images

ckSystem	System defined check images.
ckSystemFlat	Flat system defined check images.

Group

Types

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

```
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

Members	Description
csUncheckedNormal	unchecked and not pressed
csUncheckedPressed	unchecked and pressed
csCheckedNormal	checked and not pressed
csCheckedPressed	checked and pressed
csMixedNormal	3-state check box and not pressed
csMixedPressed	3-state check box and pressed

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TCheckState Enumeration
| TDragOperation Enumeration

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!](#)



Types | TCheckType Enumeration
| TVTGetImageExEvent Type

TDragOperation Enumeration

Types

Not documented.

Pascal

```
TDragOperation = (doCopy, doMove, doLink);
```

Description

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

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TDragOperation
Enumeration | TDropMode
Enumeration

TVTGetImageExEvent Type

Types

Not documented.

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!](#)



[Types](#) | [TVTGetImageExEvent
Type](#) | [THeaderState Enumeration](#)

TDropMode Enumeration

[Types](#)

Not documented.

Pascal

```
TDropMode = (dmNowhere, dmAbove, dmOnNode, dmBelow);
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



THeaderState Enumeration

Types

Not documented.

Pascal

```
THeaderState = (hsAutoSizing, hsDragging, hsDragPend
```

Description

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

Members

Members	Description
hsAutoSizing	auto size chain is in progress, do not trigger again on WM_SIZE
hsDragging	header dragging is in progress (only if enabled)
hsDragPending	left button is down, user might want to start dragging a column
hsLoading	The header currently loads from stream, so updates are not necessary.
hsTracking	column resizing is in progress
hsTrackPending	left button is down, user might want to start resize a column

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | THeaderState
Enumeration | THitPosition
Enumeration

THintAnimationType Enumeration

Types

Not documented.

Pascal

```
THintAnimationType = (hatNone, hatFade, hatSlide, ha
```

Description

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

Members

Members	Description
hatNone	no animation at all, just display hint/tooltip
hatFade	fade in the hint/tooltip, like in Windows 2000
hatSlide	slide in the hint/tooltip, like in Windows 98
hatSystemDefault	use what the system is using (slide for Win9x, slide/fade for Win2K+, depends on settings)

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



THitPosition Enumeration

Types

Not documented.

Pascal

```
THitPosition = (hiAbove, hiBelow, hiNowhere, hiOnItem)
```

Description

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

Members

Members	Description
hiAbove	above the client area (if relative) or the absolute tree area
hiBelow	below the client area (if relative) or the absolute tree area
hiNowhere	no node is involved (possible only if the tree is not as tall as the client area)
hiOnItem	on the bitmaps/buttons or label associated with an item
hiOnItemButton	on the button associated with an item
hiOnItemCheckbox	on the checkbox if enabled
hiOnItemIndent	in the indentation area in front of a node

hiOnItemLabel	on the normal text area associated with an item
hiOnItemLeft	when right aligned or centered)
hiOnItemRight	if left aligned or centered)
hiOnNormalIcon	on the "normal" image
hiOnStateIcon	on the state image
hiToLeft	to the left of the client area (if relative) or the absolute tree area
hiToRight	to the right of the client area (if relative) or the absolute tree area

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | [THitPosition Enumeration](#) | [TScrollBarStyle Enumeration](#)

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

Members	Description
eaColor	Use the provided color to erase the background instead the one of the tree.
eaDefault	The tree should erase the item's background (bitmap or solid).
eaNone	Let the application paint the background.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TItemEraseAction
Enumeration](#) | [TSortDirection
Enumeration](#)

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!](#)



[Types](#) | [TScrollBarStyle Enumeration](#) | [TVirtualNodeInitState Enumeration](#)

TSortDirection Enumeration

[Types](#)

Not documented.

Pascal

```
TSortDirection = (sdAscending, sdDescending);
```

Description

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

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



Types | TSortDirection
Enumeration | TVirtualNodeState
Enumeration

TVirtualNodeInitState Enumeration

Types

Not documented.

Pascal

```
TVirtualNodeInitState = (ivsDisabled, ivsExpanded, i
```

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, vsCu
```

Description

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

Members

Members	Description
vsInitialized	Set after the node has been initialized.
vsChecking	Node's check state is changing, avoid propagation.
vsCutOrCopy	Node is selected as cut or copy and paste source.
vsDisabled	Set if node is disabled.
vsDeleting	Set when the node is about to be freed.
vsExpanded	Set if the node is expanded.
vsHasChildren	Indicates the presence of child nodes without actually setting them.

<code>vsVisible</code>	Indicate whether the node is visible or not (independent of the expand states of its parents).
<code>vsSelected</code>	Set if the node is in the current selection.
<code>vsInitialUserData</code>	Set if (via <code>AddChild</code> or <code>InsertNode</code>) initial user data has been set which requires <code>OnFreeNode</code> .
<code>vsAllChildrenHidden</code>	Set if <code>vsHasChildren</code> is set and no child node has the <code>vsVisible</code> flag set.
<code>vsClearing</code>	Don't register structure change event.
<code>vsMultiline</code>	Node text is wrapped at the cell boundaries instead of being shorted.
<code>vsHeightMeasured</code>	Node height has been determined and does not need a recalculation.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVirtualNodeState
Enumeration |
TVSTTextSourceType
Enumeration

TVirtualTreeColumnStyle Enumeration

Types

Not documented.

Pascal

```
TVirtualTreeColumnStyle = (vsText, vsOwnerDraw);
```

Description

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

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TVSTTextSourceType Enumeration

Types

Not documented.

Pascal

```
TVSTTextSourceType = (tstAll, tstInitialized, tstSel
```

Description

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

Members

Members	Description
tstAll	Initialization is done on the fly.
tstInitialized	Only initialized nodes are rendered.
tstSelected	Only selected nodes are rendered.
tstCutCopySet	Only nodes currently marked as being in the cut/copy clipboard set are rendered.
tstVisible	Only visible nodes are rendered.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



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

Members	Description
ttNormal	normal label of the node, this is also the text which can be edited
ttStatic	static (non-editable) text after the normal text

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVSTTextType
Enumeration | TVTAutoOption
Enumeration

TVTAnimationOption Enumeration

Types

Not documented.

Pascal

```
TVTAnimationOption = (toAnimatedToggle);
```

Description

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

Members

Members	Description
toAnimatedToggle	Expanding and collapsing a node is animated (quick window scroll).

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TVTAutoOption Enumeration

Types

Not documented.

Pascal

```
TVTAutoOption = (toAutoDropExpand, toAutoExpand, toA
```

Description

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

Members

Members	Description
toAutoDropExpand	Expand node if it is the drop target for more than certain time.
toAutoExpand	Nodes are expanded (collapsed) when getting (losing) the focus.
toAutoScroll	Scroll if mouse is near the border while dragging or selecting.
toAutoScrollOnExpand	Scroll as many child nodes in view as possible after expanding a node.
toAutoSort	Sort tree when Header.SortColumn or Header.SortDirection change

	or sort node if child nodes are added.
toAutoSpanColumns	Large entries continue into next column(s) if there's no text in them (no clipping).
toAutoTristateTracking	Checkstates are automatically propagated for tri state check boxes.
toAutoHideButtons	Node buttons are hidden when there are child nodes, but all are invisible.
toAutoDeleteMovedNodes	Delete nodes which where moved in a drag operation (if not directed otherwise).
toDisableAutoscrollOnFocus	Disable scrolling a column entirely into view if it gets focused.
toAutoChangeScale	Change default node height automatically if the system's font scale is set to big fonts.
toAutoFreeOnCollapse	Frees any child node after a node has been collapsed (HasChildren flag stays there).

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTAutoOption
Enumeration | TVTButtonStyle
Enumeration

TVTButtonFillMode Enumeration

Types

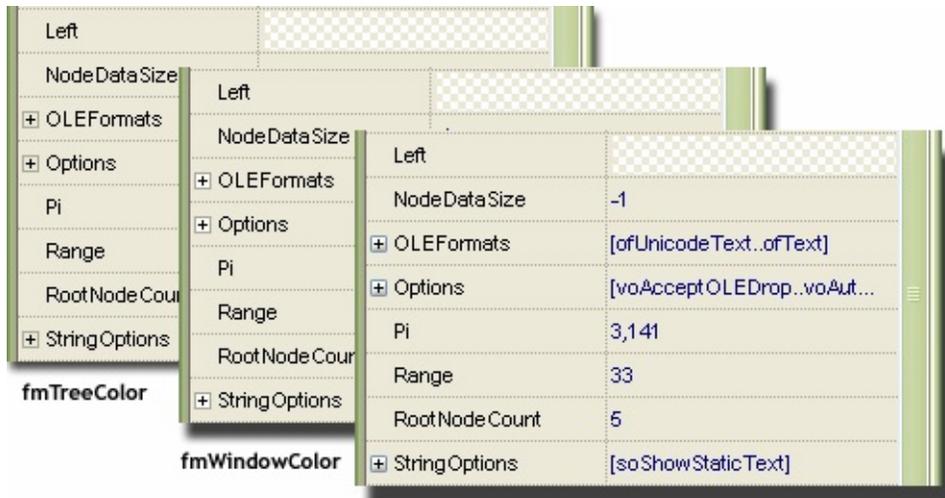
Determines how the interior of nodes buttons should be drawn.

Pascal

```
TVTButtonFillMode = (fmTreeColor, fmWindowColor, fmS
```

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:



theme aware or fmShaded for legacy code

Members

Members	Description
fmTreeColor	solid color, uses the tree's background color
fmWindowColor	solid color, uses clWindow
fmShaded	color gradient, Windows XP style (legacy code, use toThemeAware on Windows XP instead)
fmTransparent	transparent color, use the item's background color

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTButtonFillMode
Enumeration | TVTColumnOption
Enumeration

TVTButtonStyle Enumeration

Types

Not documented.

Pascal

```
TVTButtonStyle = (bsRectangle, bsTriangle);
```

Description

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

Members

Members	Description
bsRectangle	traditional Windows look (plus/minus buttons)
bsTriangle	traditional Macintosh look

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTButtonStyle
Enumeration | TVTDragImageKind
Enumeration

TVTColumnOption Enumeration

Types

Not documented.

Pascal

```
TVTColumnOption = (coAllowClick, coDraggable, coEnab
```

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!](#)



Types | [TVTColumnOption Enumeration](#) | [TVTDragMoveRestriction Enumeration](#)

TVTDragImageKind Enumeration

Types

Not documented.

Pascal

```
TVTDragImageKind = (diComplete, diMainColumnOnly, di
```

Description

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

Members

Members	Description
diComplete	show a complete drag image with all columns, only visible columns are shown
diMainColumnOnly	show only the main column (the tree column)
diNoImage	don't show a drag image at all

Group

[Types](#)

File

VirtualTrees

Links

[Types](#)

What do you think about this topic? [Send feedback!](#)



Types | TVTDragImageKind
Enumeration | TVTDragType
Enumeration

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!](#)



Types | TVTDragMoveRestriction
Enumeration |
TVTDrawSelectionMode
Enumeration

TVTDragType Enumeration

Types

Not documented.

Pascal

```
TVTDragType = (dtOLE, dtVCL);
```

Description

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

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTDragType
Enumeration | TVTDropMarkMode
Enumeration

TVTDrawSelectionMode Enumeration

Types

Not documented.

Pascal

```
TVTDrawSelectionMode = (smDottedRectangle, smBlended
```

Description

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

Members

Members	Description
smDottedRectangle	same as DrawFocusRect
smBlendedRectangle	alpha blending, uses special colors (see TVTColors)

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTDrawSelectionMode
Enumeration |
TVTHeaderColumnLayout
Enumeration

TVTDropMarkMode Enumeration

Types

Not documented.

Pascal

```
TVTDropMarkMode = (dmmNone, dmmLeft, dmmRight);
```

Description

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

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTDropMarkMode
Enumeration | TVTHeaderOption
Enumeration

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!](#)



Types | [TVTHeaderColumnLayout Enumeration](#) | [TVTHeaderPopupOption Enumeration](#)

TVTHeaderOption Enumeration

Types

Not documented.

Pascal

```
TVTHeaderOption = (hoAutoSize, hoColumnResize, hoD
```

Description

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

Members

Members	Description
hoAutoSize	adjust a column so that the header never exceeds client width of owner control
hoColumnResize	resizing columns is allowed
hoDb1ClickResize	allows a column to resize itself to its largest entry
hoDrag	dragging columns is allowed
hoHotTrack	header captions are highlighted when mouse is over a particular column
hoOwnerDraw	header items with the owner draw style can be drawn by the application via event

hoRestrictDrag	header can only be dragged horizontally
hoShowHint	show application defined header hint
hoShowImages	show images
hoShowSortGlyphs	allow visible sort glyphs
hoVisible	header is visible

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TVTHeaderPopupOption Enumeration

Types

Not documented.

Pascal

```
TVTHeaderPopupOption = (poOriginalOrder, poAllowHide
```

Description

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

Members

Members	Description
poOriginalOrder	Show menu items in original column order as they were added to the tree.
poAllowHideAll	Allows to hide all columns, including the last one.

Group

Types

File

VTHeaderPopup

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTHeaderPopupOption
Enumeration | TVTHeaderStyle
Enumeration

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

VTHeaderPopup

Links

Types

What do you think about this topic? [Send feedback!](#)



TVTHeaderStyle Enumeration

Types

Not documented.

Pascal

```
TVTHeaderStyle = (hsThickButtons, hsFlatButtons, hsP
```

Description

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

Members

Members	Description
hsThickButtons	TButton look and feel
hsFlatButtons	flatter look than hsThickButton, like an always raised flat TToolButton
hsPlates	flat TToolButton look and feel (raise on hover etc.)
hsXPStyle	Windows XP style

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TVTHintMode Enumeration

Types

Not documented.

Pascal

```
TVTHintMode = (hmDefault, hmHint, hmHintAndDefault,
```

Description

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

Members

Members	Description
hmDefault	show the hint of the control
hmHint	show node specific hint string returned by the application
hmHintAndDefault	same as hmHint but show the control's hint if no node is concerned
hmTooltip	show the text of the node if it isn't already fully shown

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTHintMode
Enumeration | TVTImageKind
Enumeration

TVTImageInfoIndex Enumeration

Types

Not documented.

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!](#)



[Types](#) | [TVTImageInfoIndex Enumeration](#) | [TVTIncrementalSearch Enumeration](#)

TVTImageKind Enumeration

Types

Not documented.

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!](#)



Types | [TVTImageKind Enumeration](#) | [TVTInternalPaintOption Enumeration](#)

TVTIncrementalSearch Enumeration

Types

Not documented.

Pascal

```
TVTIncrementalSearch = (isAll, isNone, isInitialized
```

Description

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

Members

Members	Description
isAll	search every node in tree, initialize if necessary
isNone	disable incremental search
isInitializedOnly	search only initialized nodes, skip others
isVisibleOnly	search only visible nodes, initialize if necessary

Group

[Types](#)

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | [TVTIncrementalSearch Enumeration](#) | [TVTLineMode Enumeration](#)

TVTInternalPaintOption Enumeration

Types

Not documented.

Pascal

```
TVTInternalPaintOption = (poBackground, poColumnColor)
```

Description

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

Members

Members	Description
poBackground	draw background image if there is any and it is enabled
poColumnColor	erase node's background with the column's color
poDrawFocusRect	draw focus rectangle around the focused node
poDrawSelection	draw selected nodes with the normal selection color
poDrawDropMark	draw drop mark if a node is currently the drop target
poGridLines	draw grid lines if enabled
poMainOnly	draw only the main column
poSelectedOnly	draw only selected nodes

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTInternalPaintOption
Enumeration | TVTLineStyle
Enumeration

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

Members	Description
lmNormal	usual tree lines (as in TTreeView)
lmBands	looks similar to a Nassi-Schneidermann diagram

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TVTLineStyle Enumeration

Types

Not documented.

Pascal

```
TVTLineStyle = (lsCustomStyle, lsDotted, lsSolid);
```

Description

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

Members

Members	Description
lsCustomStyle	application provides a line pattern
lsDotted	usual dotted lines (default)
lsSolid	simple solid lines

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TVTLineType Enumeration

Types

Not documented.

Pascal

```
TVTLineType = (ltNone, ltBottomRight, ltTopDown, ltT
```

Description

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

Members

Members	Description
ltNone	no line at all
ltBottomRight	a line from bottom to the center and from there to the right
ltTopDown	a line from top to bottom
ltTopDownRight	a line from top to bottom and from center to the right
ltRight	a line from center to the right
ltTopRight	a line from bottom to center and from there to the right special styles for alternative drawings of tree lines
ltLeft	a line from top to bottom at the left

ItLeftBottom

a combination of ItLeft and a line
at the bottom from left to right

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TVTMiscOption Enumeration

Types

Not documented.

Pascal

```
TVTMiscOption = (toAcceptOLEDrop, toCheckSupport, to
```

Description

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

Members

Members	Description
toAcceptOLEDrop	Register tree as OLE accepting drop target
toCheckSupport	Show checkboxes/radio buttons.
toEditable	Node captions can be edited.
toFullRepaintOnResize	Fully invalidate the tree when its window is resized (CS_HREDRAW/CS_VREDRAW).
toGridExtensions	Use some special enhancements to simulate and support grid behavior.
toInitOnSave	Initialize nodes when saving a tree to a stream.
toReportMode	Tree behaves like TListView in report mode.

toToggleOnDb1Click	Toggle node expansion state when it is double clicked.
toWheelPanning	This option and toMiddleClickSelect are mutal exclusive, where panning has precedence.
toReadOnly	No action is executed and node editing is not possible.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Types | TVTMiscOption
Enumeration |
TVTNodeAttachMode
Enumeration

TVTNodeAlignment Enumeration

Types

Not documented.

Pascal

```
TVTNodeAlignment = (naFromBottom, naFromTop, naPropo
```

Description

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

Members

Members	Description
naFromBottom	the align member specifies amount of units (usually pixels) from top border of the node
naFromTop	align is to be measured from bottom
naProportional	align is to be measure in percent of the entire node height and relative to top

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TVTNodeAttachMode Enumeration

Types

Not documented.

Pascal

```
TVTNodeAttachMode = (amNowhere, amInsertBefore, amIn
```

Description

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

Members

Members	Description
amNowhere	just for simplified tests, means to ignore the Add/Insert command
amInsertBefore	insert node just before destination (as sibling of destination)
amInsertAfter	insert node just after destination (as sibling of destination)
amAddChildFirst	add node as first child of destination
amAddChildLast	add node as last child of destination

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



[Types](#) | [TVTNodeAttachMode Enumeration](#) | [TVTPaintOption Enumeration](#)

TVTScrollbarShowEvent Type

[Types](#)

Not documented.

Pascal

```
TVTScrollbarShowEvent = 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!](#)



Types | TVTScrollbarShowEvent
Type | TVTSearchDirection
Enumeration

TVTPaintOption Enumeration

Types

Not documented.

Pascal

```
TVTPaintOption = (toHideFocusRect, toHideSelection,
```

Description

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

Members

Members	Description
toHideFocusRect	Avoid drawing the dotted rectangle around the currently focused node.
toHideSelection	Selected nodes are drawn as unselected nodes if the tree is unfocused.
toHotTrack	Track which node is under the mouse cursor.
toPopupMode	Paint tree as would it always have the focus (useful for tree combo boxes etc.)
toShowBackground	Use the background image if there's one.
	Display collapse/expand buttons

toShowButtons	left to a node.
toShowDropmark	Show the dropmark during drag'n drop operations.
toShowHorzGridLines	Display horizontal lines to simulate a grid.
toShowRoot	Show lines also at top level (does not show the hidden/internal root node).
toShowTreeLines	Display tree lines to show hierarchy of nodes.
toShowVertGridLines	Display vertical lines (depending on columns) to simulate a grid.
toThemeAware	Draw UI elements (header, tree buttons etc.) according to the current theme if enabled (Windows XP+ only, application must be themed).
toUseBlendedImages	Enable alpha blending for ghosted nodes or those which are being cut/copied.
toGhostedIfUnfocused	Ghosted images are still shown as ghosted if unfocused (otherwise they become non-ghosted images).
toFullVertGridLines	This option only has an effect if toShowVertGridLines is enabled too.
toAlwaysHideSelection	Do not draw node selection, regardless of focused state.
toUseBlendedSelection	Enable alpha blending for node selections.

Group
Types

File
VirtualTrees

Links
Types

What do you think about this topic? [Send feedback!](#)



Types | TVTPaintOption
Enumeration | TVTSearchStart
Enumeration

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!](#)



Types | TVTSearchDirection
Enumeration | TVTSelectionOption
Enumeration

TVTSearchStart Enumeration

Types

Not documented.

Pascal

```
TVTSearchStart = (ssAlwaysStartOver, ssLastHit, ssFo
```

Description

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

Members

Members	Description
ssAlwaysStartOver	always use the first/last node (depending on direction) to search from
ssLastHit	use the last found node
ssFocusedNode	use the currently focused node

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TVTSelectionOption Enumeration

Types

Not documented.

Pascal

```
TVTSelectionOption = (toDisableDrawSelection, toExte
```

Description

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

Members

Members	Description
<code>toDisableDrawSelection</code>	Prevent user from selecting with the selection rectangle in multiselect mode.
<code>toExtendedFocus</code>	Entries other than in the main column can be selected, edited etc.
<code>toFullRowSelect</code>	Hit test as well as selection highlight are not constrained to the text of a node.
<code>toLevelSelectConstraint</code>	Constrain selection to the same level as the selection anchor.
<code>toMiddleClickSelect</code>	with the middle mouse button. This and <code>toWheelPanning</code> are mutual exclusive.

<code>toMultiSelect</code>	Allow more than one node to be selected.
<code>toRightClickSelect</code>	with the right mouse button.
<code>toSiblingSelectConstraint</code>	constrain selection to nodes with same parent
<code>toCenterScrollIntoView</code>	Center nodes vertically in the client area when scrolling into view.
<code>toSimpleDrawSelection</code>	Simplifies draw selection, so a node's caption does not need to intersect with the selection rectangle.

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

Members	Description
toSaveCaptions	If set then the caption is automatically saved with the tree node, regardless of what is saved in the user data.
toShowStaticText	Show static text in a caption which can be differently formatted than the caption but cannot be edited.
toAutoAcceptEditChange	If not set then changes are cancelled.

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



TVTUpdateState Enumeration

Types

Not documented.

Pascal

```
TVTUpdateState = (usBegin, usBeginSynch, usSynch, us
```

Description

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

Members

Members	Description
usBegin	The tree just entered the update state (BeginUpdate call for the first time).
usBeginSynch	The tree just entered the synch update state (BeginSynch call for the first time).
usSynch	Begin/EndSynch has been called but the tree did not change the update state.
usUpdate	Begin/EndUpdate has been called but the tree did not change the update state.
usEnd	The tree just left the update state (EndUpdate called for the last level).

usEndSynch

The tree just left the synch update state (EndSynch called for the last level).

Group

Types

File

VirtualTrees

Links

Types

What do you think about this topic? [Send feedback!](#)



Variables

[Virtual Treeview](#) | [Variables](#) | [Legend](#)

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_VRTFNOOBS](#)
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.
- ◆ **NeedToUnitalize**
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!](#)



[Variables](#) | [CF_CSV Variable](#) | [CF_VIRTUALTREE Variable](#)

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!](#)



[Variables](#) | [CF_HTML Variable](#) |
[CF_VRTF Variable](#)

CF_VIRTUALTREE Variable

[Variables](#)

Not documented.

Pascal

```
CF_VIRTUALTREE: Word;
```

Description

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

Group

[Variables](#)

File

VirtualTrees

Links

[Variables](#)

What do you think about this topic? [Send feedback!](#)



Variables | [CF_VIRTUALTREE](#)
Variable | [CF_VRTFNOOBS](#)
Variable

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!](#)



[Variables](#) | [CF_VRTF Variable](#) |
[CF_VTREFERENCE Variable](#)

CF_VRTFNOOBS Variable

[Variables](#)

Not documented.

Pascal

```
CF_VRTFNOOBS: 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!](#)



Variables | [CF_VRTFNOOBS](#)
Variable | [ClipboardDescriptions](#)
Variable

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!](#)



Variables | [CF_VTREFERENCE](#)
Variable | [DarkCheckImages](#)
Variable

ClipboardDescriptions Variable

[Variables](#)

Not documented.

Pascal

```
ClipboardDescriptions: array [1..CF_MAX - 1] of TCl
```

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!](#)



[Variables](#) | [ClipboardDescriptions Variable](#) | [DarkTickImages Variable](#)

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!](#)



[Variables](#) | [DarkCheckImages Variable](#) | [FlatImages Variable](#)

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!](#)



[Variables](#) | [DarkTickImages Variable](#) | [HintFont Variable](#)

FlatImages Variable

[Variables](#)

Not documented.

Pascal

```
FlatImages: TImageList;
```

Description

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

Group

[Variables](#)

File

VirtualTrees

Links

[Variables](#)

What do you think about this topic? [Send feedback!](#)



[Variables](#) | [FlatImages Variable](#) | [HintWindowDestroyed Variable](#)

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!](#)



[Variables](#) | [HintFont Variable](#) | [Initialized Variable](#)

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!](#)



[Variables](#) | [HintWindowDestroyed
Variable](#) |
[InternalClipboardFormats
Variable](#)

Initialized Variable

[Variables](#)

Not documented.

Pascal

```
Initialized: Boolean;
```

Description

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

Group

[Variables](#)

File

VirtualTrees

Links

[Variables](#)

What do you think about this topic? [Send feedback!](#)



[Variables](#) | [Initialized Variable](#) |
[IsWin2K Variable](#)

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!](#)



[Variables](#) |
[InternalClipboardFormats Variable](#)
| [IsWinNT Variable](#)

IsWin2K Variable

[Variables](#)

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!](#)



[Variables](#) | [IsWin2K Variable](#) | [IsWinXP Variable](#)

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!](#)



[Variables](#) | [IsWinNT Variable](#) | [LightCheckImages Variable](#)

IsWinXP Variable

[Variables](#)

Not documented.

Pascal

```
IsWinXP: Boolean;
```

Description

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

Group

[Variables](#)

File

VirtualTrees

Links

[Variables](#)

What do you think about this topic? [Send feedback!](#)



[Variables](#) | [IsWinXP Variable](#) | [LightTickImages Variable](#)

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!](#)



[Variables](#) | [LightCheckImages Variable](#) | [MMXAvailable Variable](#)

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!](#)



[Variables](#) | [LightTickImages](#)
[Variable](#) | [NeedToUnitalize](#)
[Variable](#)

MMXAvailable Variable

[Variables](#)

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!](#)



[Variables](#) | [MMXAvailable Variable](#)
| [StandardOLEFormat Variable](#)

NeedToUnitialize Variable

[Variables](#)

Not documented.

Pascal

```
NeedToUnitialize: Boolean;
```

Description

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

Group

[Variables](#)

File

VirtualTrees

Links

[Variables](#)

What do you think about this topic? [Send feedback!](#)



[Variables](#) | [NeedToUnitalizeVariable](#) | [SystemCheckImagesVariable](#)

StandardOLEFormat Variable

[Variables](#)

Not documented.

Pascal

```
StandardOLEFormat: TFormatEtc = ( cfFormat: 0; ptd:
```

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!](#)



[Variables](#) | [StandardOLEFormat
Variable](#) |
[SystemFlatCheckImages Variable](#)

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!](#)



[Variables](#) | [SystemCheckImages Variable](#) | [UtilityImages Variable](#)

SystemFlatCheckImages Variable

[Variables](#)

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!](#)



[Variables](#) |
[SystemFlatCheckImages Variable](#)
| [Watcher Variable](#)

UtilityImages Variable

[Variables](#)

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!](#)



[Variables](#) | [UtilityImages Variable](#) | [WorkerThread Variable](#)

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!](#)



[Variables](#) | [Watcher Variable](#) |
[WorkEvent Variable](#)

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!](#)



[Variables](#) | [WorkerThread Variable](#)
| [XPImages Variable](#)

WorkEvent Variable

[Variables](#)

Not documented.

Pascal

```
WorkEvent: THandle;
```

Description

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

Group

[Variables](#)

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!](#)



Constants

[Constants](#) | [Virtual Treeview](#) | [Topics](#) | [Legend](#)

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**

◆ **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.
- ❖ **VTHeaderStreamVersion**
Not documented.
- ❖ **VTTreeStreamVersion**
Not documented.
- ❖ **VTVersion**
Not documented.
- ❖ **WideCR**
Not documented.
- ❖ **WideLF**
Not documented.
- ❖ **WideLineSeparator**
Not documented.
- ❖ **WideNull**
Not documented.
- ❖ **WM_CHANGE_STATE**
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!](#)



[Constants](#) | [AlignmentToDrawFlag Constant](#) | [BaseChunk Constant](#)

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!](#)



[Constants](#) | [AllocIncrement
Constant](#) | [CacheThreshold
Constant](#)

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!](#)



[Constants](#) | [BaseChunk Constant](#) | [CaptionChunk Constant](#)

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!](#)



[Constants](#) | [CacheThreshold Constant](#) | [CFSTR_CSV Constant](#)

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

Constants

Contains the registration string for certain clipboard formats.

Pascal

```
CFSTR_VIRTUALTREE = 'Virtual Tree Data';  
CFSTR_VTREFERENCE = 'Virtual Tree Reference';  
CFSTR_HTML = 'HTML Format';  
CFSTR_RTF = 'Rich Text Format';  
CFSTR_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!](#)



[Constants](#) | [CFSTR_CSV](#)
[Constant](#) | [Check button image](#)
[indices](#)

ChangeTimer Constant

[Constants](#)

Not documented.

Pascal

```
ChangeTimer = 5;
```

Description

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

Group

[Constants](#)

File

VirtualTrees

Links

[Constants](#)

What do you think about this topic? [Send feedback!](#)



[Constants](#) | [ChangeTimer](#)
[Constant](#) | [ClipboardStates](#)
[Constant](#)

Check button image indices

[Constants](#)

[Constants](#)

Links

[Constants](#)

What do you think about this topic? [Send feedback!](#)



[Constants](#) | [Check button image indices](#) | [CLSID_DragDropHelper Constant](#)

ClipboardStates Constant

[Constants](#)

Not documented.

Pascal

```
ClipboardStates = [tsCopyPending, tsCutPending];
```

Description

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

Group

[Constants](#)

File

VirtualTrees

Links

[Constants](#)

What do you think about this topic? [Send feedback!](#)



[Constants](#) | [ClipboardStates](#)
[Constant](#) | [CM_AUTOADJUST](#)
[Constant](#)

CLSID_DragDropHelper Constant

[Constants](#)

Not documented.

Pascal

```
CLSID_DragDropHelper: TGUID = (D1: $4657278A; D2: $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!](#)



CM_AUTOADJUST Constant

Constants

Not documented.

Pascal

```
CM_AUTOADJUST = CM_BASE + 2005;
```

Description

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

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? [Send feedback!](#)



CM_DENYSUBCLASSING Constant

Constants

Not documented.

Pascal

```
CM_DENYSUBCLASSING = CM_BASE + 2000;
```

Description

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

Group

[Constants](#)

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 M
```

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

```
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

```
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

Constants

Not documented.

Pascal

```
DefaultAutoOptions = [toAutoDropExpand, toAutoTrista
```

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!](#)



Constants | DefaultAutoOptions
Constant | DefaultMiscOptions
Constant

DefaultColumnOptions Constant

Constants

Not documented.

Pascal

```
DefaultColumnOptions = [coAllowClick, coDraggable, c
```

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!](#)



[Constants](#) | [DefaultColumnOptions
Constant](#) | [DefaultPaintOptions
Constant](#)

DefaultMiscOptions Constant

[Constants](#)

Not documented.

Pascal

```
DefaultMiscOptions = [toAcceptOLEDrop, toFullRepaint
```

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!](#)



[Constants](#) | [DefaultMiscOptions
Constant](#) |
[DefaultScrollUpdateFlags
Constant](#)

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!](#)



[Constants](#) | [DefaultPaintOptions
Constant](#) |
[DefaultSelectionOptions Constant](#)

DefaultScrollUpdateFlags Constant

[Constants](#)

Not documented.

Pascal

```
DefaultScrollUpdateFlags = [suoRepaintHeader, suoRep
```

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

Constants

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!](#)



[Constants](#) | [DefaultStringOptions Constant](#) | [ExpandTimer Constant](#)

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!](#)



[Constants](#) | [EditTimer Constant](#) |
[FadeAnimationStepCount](#)
[Constant](#)

ExpandTimer Constant

[Constants](#)

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!](#)



[Constants](#) | [ExpandTimer Constant](#)
| [Grays Constant](#)

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!](#)



[Constants](#) |
[FadeAnimationStepCount](#)
[Constant](#) |
[hcTFCannotSetUserData](#) [Constant](#)

Grays Constant

[Constants](#)

Not documented.

Pascal

```
Grays: array[0..3] of TColor = (clWhite, clSilver, c
```

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!](#)



[Constants](#) | [Grays Constant](#) | [hcTFClipboardFailed Constant](#)

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

Constants

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!](#)



[Constants](#) | [hcTFClipboardFailed
Constant](#) | [hcTFCorruptStream2
Constant](#)

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](#)

What do you think about this topic? [Send feedback!](#)



[Constants](#) | [hcTFCorruptStream1
Constant](#) | [hcTFEditLinkIsNil
Constant](#)

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!](#)



Constants | [hcTFCorruptStream2
Constant](#) | [hcTFStreamTooSmall
Constant](#)

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!](#)



[Constants](#) | [hcTFEditLinkIsNil
Constant](#) | [hcTFWrongMoveError
Constant](#)

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!](#)



Constants | [hcTFStreamTooSmall](#)
Constant |
[hcTFWrongStreamFormat](#)
Constant

hcTFWrongMoveError Constant

[Constants](#)

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!](#)



Constants | [hcTFWrongMoveError](#)
Constant |
[hcTFWrongStreamVersion](#)
Constant

hcTFWrongStreamFormat Constant

[Constants](#)

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

Constants

Not documented.

Pascal

```
HeaderTimer = 3;
```

Description

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

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? [Send feedback!](#)



[Constants](#) | [HeaderTimer Constant](#)
| [IID_IDropTarget Constant](#)

IID_IDragSourceHelper Constant

[Constants](#)

Not documented.

Pascal

```
IID_IDragSourceHelper: TGUID = (D1: $DE5BF786; D2: $
```

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!](#)



[Constants](#) |
[IID_IDragSourceHelper Constant](#) |
[IID_IDropTargetHelper Constant](#)

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!](#)



[Constants](#) | [IID_IDropTarget Constant](#) | [InvalidColumn Constant](#)

IID_IDropTargetHelper Constant

[Constants](#)

Not documented.

Pascal

```
IID_IDropTargetHelper: TGUID = (D1: $4657278B; D2: $
```

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!](#)



[Constants](#) | [IID_IDropTargetHelper Constant](#) | [MagicID Constant](#)

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!](#)



[Constants](#) | [InvalidColumnConstant](#) | [MinimumTimerIntervalConstant](#)

MagicID Constant

[Constants](#)

Not documented.

Pascal

```
MagicID: TMagicID = (#$2045, 'V', 'T', WideChar(VTTr
```

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!](#)



[Constants](#) | [MagicID Constant](#) |
[MouseButtonDown Constant](#)

MinimumTimeInterval Constant

[Constants](#)

Not documented.

Pascal

```
MinimumTimeInterval = 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!](#)



[Constants](#) | [MinimumTimerInterval Constant](#) | [NoColumn Constant](#)

MouseButtonDown Constant

[Constants](#)

Not documented.

Pascal

```
MouseButtonDown = [tsLeftButtonDown, tsMiddleButtonD
```

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!](#)



[Constants](#) | [MouseButtonDown Constant](#) | [NodeChunk Constant](#)

NoColumn Constant

[Constants](#)

Not documented.

Pascal

```
NoColumn = -1;
```

Description

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

Group

[Constants](#)

File

VirtualTrees

Links

[Constants](#)

What do you think about this topic? [Send feedback!](#)



[Constants](#) | [NoColumn Constant](#) | [OptionMap Constant](#)

NodeChunk Constant

[Constants](#)

Not documented.

Pascal

```
NodeChunk = 1;
```

Description

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

Group

[Constants](#)

File

VirtualTrees

Links

[Constants](#)

What do you think about this topic? [Send feedback!](#)



[Constants](#) | [NodeChunk Constant](#) | [PressedState Constant](#)

OptionMap Constant

[Constants](#)

Not documented.

Pascal

```
OptionMap: array[TOldVTOption] of Integer = ( Ord(to
```

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

```
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!](#)



[Constants](#) | [PressedState
Constant](#) | [SCannotSetUserData
Constant](#)

RTLFlag Constant

[Constants](#)

Not documented.

Pascal

```
RTLFlag: array[Boolean] of Integer = (0, ETO_RTLEAD
```

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!](#)



[Constants](#) | [RTLFlag Constant](#) |
[SClipboardFailed Constant](#)

SCannotSetUserData Constant

[Constants](#)

Not documented.

Pascal

```
SCannotSetUserData = 'Cannot set initial user data b
```

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!](#)



[Constants](#) | [SCannotSetUserData
Constant](#) | [SCorruptStream1
Constant](#)

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!](#)



[Constants](#) | [SClipboardFailed
Constant](#) | [SCorruptStream2
Constant](#)

SCorruptStream1 Constant

[Constants](#)

Not documented.

Pascal

```
SCorruptStream1 = 'Stream data corrupt. A node''s an
```

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!](#)



[Constants](#) | [SCorruptStream1 Constant](#) | [ScrollTimer Constant](#)

SCorruptStream2 Constant

[Constants](#)

Not documented.

Pascal

```
SCorruptStream2 = 'Stream data corrupt. Unexpected d
```

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!](#)



[Constants](#) | [SCorruptStream2 Constant](#) | [SearchTimer Constant](#)

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!](#)



[Constants](#) | [ScrollTimer Constant](#) | [SEditLinkIsNil Constant](#)

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!](#)



[Constants](#) | [SearchTimer Constant](#)
| [ShadowSize Constant](#)

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!](#)



[Constants](#) | [ShadowSize Constant](#)
| [SID_IDropTarget Constant](#)

SID_IDragSourceHelper Constant

[Constants](#)

Not documented.

Pascal

```
SID_IDragSourceHelper = '{DE5BF786-477A-11D2-839D-00
```

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!](#)



[Constants](#) |
[SID_IDragSourceHelper Constant](#)
| [SID_IDropTargetHelper Constant](#)

SID_IDropTarget Constant

[Constants](#)

Not documented.

Pascal

```
SID_IDropTarget = '{00000122-0000-0000-C000-00000000}
```

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!](#)



[Constants](#) | [SID_IDropTarget
Constant](#) | [SStreamTooSmall
Constant](#)

SID_IDropTargetHelper Constant

[Constants](#)

Not documented.

Pascal

```
SID_IDropTargetHelper = '{4657278B-411B-11D2-839A-00
```

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!](#)



[Constants](#) |
[SID_IDropTargetHelper Constant](#) |
[StructureChangeTimer Constant](#)

SStreamTooSmall Constant

[Constants](#)

Not documented.

Pascal

```
SStreamTooSmall = 'Unable to load tree structure, no
```

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!](#)



[Constants](#) | [SStreamTooSmall
Constant](#) | [SWrongMoveError
Constant](#)

StructureChangeTimer Constant

[Constants](#)

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!](#)



Constants | StructureChangeTimer
Constant | SWrongStreamFormat
Constant

SWrongMoveError Constant

Constants

Not documented.

Pascal

```
SWrongMoveError = 'Target node cannot be a child nod
```

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!](#)



[Constants](#) | [SWrongMoveError
Constant](#) | [SWrongStreamVersion
Constant](#)

SWrongStreamFormat Constant

[Constants](#)

Not documented.

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!](#)



[Constants](#) | [SWrongStreamFormat Constant](#) | [SysGrays Constant](#)

SWrongStreamVersion Constant

[Constants](#)

Not documented.

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!](#)



[Constants](#) | [SWrongStreamVersion Constant](#) | [TreeNodeSize Constant](#)

SysGrays Constant

[Constants](#)

Not documented.

Pascal

```
SysGrays: array[0..3] of TColor = (clWindow, clBtnFa
```

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!](#)



Constants | UnpressedState
Constant | UtilityImageSize
Constant

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!](#)



[Constants](#) | [UserChunk Constant](#) | [VTHeaderStreamVersion Constant](#)

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!](#)



[Constants](#) | [UtilityImageSize
Constant](#) | [VTTreeStreamVersion
Constant](#)

VTHdrStreamVersion Constant

[Constants](#)

Not documented.

Pascal

```
VTHdrStreamVersion = 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!](#)



[Constants](#) | [VTTStreamVersion Constant](#) | [WideCR Constant](#)

VTVersion Constant

[Constants](#)

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!](#)



[Constants](#) | [VTVersion Constant](#) |
[WideLF Constant](#)

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!](#)



[Constants](#) | [WideCR Constant](#) |
[WideLineSeparator Constant](#)

WideLF Constant

[Constants](#)

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!](#)



[Constants](#) | [WideLF Constant](#) | [WideNull Constant](#)

WideLineSeparator Constant

[Constants](#)

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!](#)



Constants | [WideLineSeparator](#)
Constant | [WM_CHANGE_STATE](#)
Constant

WideNull Constant

Constants

Not documented.

Pascal

```
WideNull = WideChar(#0);
```

Description

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

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? [Send feedback!](#)



[Constants](#) | [WideNull Constant](#) | [XPDarkGradientColor Constant](#)

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!](#)



Constants | WM_CHANGESTATE
Constant | XPDarkSplitBarColor
Constant

XPDarkGradientColor Constant

Constants

Not documented.

Pascal

```
XPDarkGradientColor = $B8C7CB;
```

Description

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

Group

Constants

File

VirtualTrees

Links

Constants

What do you think about this topic? [Send feedback!](#)



[Constants](#) | [XPDarkGradientColor
Constant](#) | [XPDownInnerLineColor
Constant](#)

XPDarkSplitBarColor Constant

[Constants](#)

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!](#)



[Constants](#) | [XPDarkSplitBarColor Constant](#) | [XPDownMiddleLineColor Constant](#)

XPDownInnerLineColor Constant

[Constants](#)

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!](#)



[Constants](#) |
[XPDownInnerLineColor Constant](#) |
[XPDownOuterLineColor Constant](#)

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!](#)



[Constants](#) |
[XPDownMiddleLineColor Constant](#)
| [XPLightSplitBarColor Constant](#)

XPDownOuterLineColor Constant

[Constants](#)

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!](#)



[Constants](#) |
[XPDownOuterLineColor Constant](#) |
[XPMainHeaderColorDown
Constant](#)

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!](#)



Constants | [XPLightSplitBarColor](#)
Constant |
[XPMainHeaderColorHover](#)
Constant

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!](#)



[Constants](#) |
[XPMainHeaderColorDown](#)
[Constant](#) | [XPMainHeaderColorUp](#)
[Constant](#)

XPMainHeaderColorHover Constant

[Constants](#)

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

[Virtual Treeview](#) | [Interfaces](#) | [Legend](#)

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

```
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



File

VirtualTrees

Links

[Symbol Reference](#), [Methods](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



IDragSourceHelper.InitializeFromBitmap Method

IDragSourceHelper Interface

Not documented.

Pascal

```
[SID_IDragSourceHelper]  
function InitializeFromBitmap(var SHDragImage: TSHDr
```

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

```
function InitializeFromWindow(Window: HWND; var ppt:
```

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!](#)



[Symbol Reference](#) |
[IDragSourceHelper Interface](#) |
[IVTDragManager Interface](#)

IDropTargetHelper Interface

[Symbol Reference](#) | [Methods](#) | [Legend](#)

Not documented.

Pascal

```
IDropTargetHelper = interface(IUnknown);
```

Description

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

Group

[Symbol Reference](#)

Members

Methods

  [DragEnter](#)

Not documented.

  [DragLeave](#)

Not documented.

  [DragOver](#)

Not documented.

  [Drop](#)

Not documented.

  [Show](#)

Not documented.

Legend

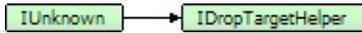


public



Method

Class Hierarchy



File

VirtualTrees

Links

[Symbol Reference](#), [Methods](#), [Legend](#)

What do you think about this topic? [Send feedback!](#)



IDropTargetHelper.DragEnter Method

IDropTargetHelper Interface

Not documented.

Pascal

```
[SID_IDropTargetHelper]  
function DragEnter(hwndTarget: HWND; pDataObject: ID
```

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 Interface](#) |
[IDropTargetHelper.DragEnter](#)
Method |
[IDropTargetHelper.DragOver](#)
Method

IDropTargetHelper.DragLeave Method

[IDropTargetHelper Interface](#)

Not documented.

Pascal

```
function DragLeave: HRESULT; stdcall;
```

Description

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

Interface

[IDropTargetHelper Interface](#)

Links

[IDropTargetHelper Interface](#)

What do you think about this topic? [Send feedback!](#)



[IDropTargetHelper Interface](#) |
[IDropTargetHelper.DragLeave
Method](#) | [IDropTargetHelper.Drop
Method](#)

IDropTargetHelper.DragOver Method

[IDropTargetHelper Interface](#)

Not documented.

Pascal

```
function DragOver(var ppt: TPoint; dwEffect: Integer
```

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 Interface](#) |
[IDropTargetHelper.DragOver
Method](#) | [IDropTargetHelper.Show
Method](#)

IDropTargetHelper.Drop Method

[IDropTargetHelper Interface](#)

Not documented.

Pascal

```
function Drop(pDataObject: IDataObject; var ppt: TPO
```

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

```
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!](#)



[Symbol Reference](#) |
[IDropTargetHelper Interface](#) |
[IVTEditLink Interface](#)

IVTDragManager Interface

[Symbol Reference](#) | [Methods](#) | [Properties](#) | [Legend](#)

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](#)

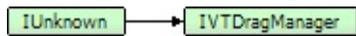
Not documented.

-   [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

```
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 Interface |
IVTDragManager.DataObject Property |
IVTDragManager.DropTargetHelperSupported
Property

IVTDragManager.DragSource Property

IVTDragManager Interface

Not documented.

Pascal

```
property DragSource: TBaseVirtualTree;
```

Description

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

Interface

[IVTDragManager Interface](#)

Links

[IVTDragManager Interface](#)

What do you think about this topic? [Send feedback!](#)



[IVTDragManager Interface](#) |
[IVTDragManager.DragSource](#)
[Property](#) |
[IVTDragManager.IsDropTarget](#)
[Property](#)

IVTDragManager.DropTargetHelperSupporto Property

[IVTDragManager Interface](#)

Not documented.

Pascal

```
property DropTargetHelperSupported: Boolean;
```

Description

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

Interface

[IVTDragManager Interface](#)

Links

[IVTDragManager Interface](#)

What do you think about this topic? [Send feedback!](#)



IVTDragManager Interface |
IVTDragManager.DropTargetHelperSupported
Property | IVTDragManager.ForceDragLeave
Method

IVTDragManager.IsDropTarget Property

IVTDragManager Interface

Not documented.

Pascal

```
property IsDropTarget: Boolean;
```

Description

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

Interface

[IVTDragManager Interface](#)

Links

[IVTDragManager Interface](#)

What do you think about this topic? [Send feedback!](#)



[IVTDragManager Interface](#) |
[IVTDragManager.IsDropTarget](#)
[Property](#) |
[IVTDragManager.GetDataObject](#)
[Method](#)

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 Interface](#) |
[IVTDragManager.ForceDragLeave](#)
Method |
[IVTDragManager.GetDragSource](#)
Method

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 Interface](#) |
[IVTDragManager.GetDataObject Method](#) |
[IVTDragManager.GetDropTargetHelperSupported Method](#)

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 Interface](#) |
[IVTDragManager.GetDragSource](#)
Method |
[IVTDragManager.GetIsDropTarget](#)
Method

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

[Symbol Reference](#) | [Methods](#) | [Legend](#)

Interface which is used for communication between the treeview and a node editor.

Pascal

```
IVTEditLink = interface;
```

Description

Due to the virtual nature of the tree it is necessary to supply a kind of plug in interface for application defined node editors. [TCustomVirtualStringTree](#) 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

[IVTEditLink](#)

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

```
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

IVTEditLink Interface

This function will be called by the virtual tree when the current editing is about to be cancelled.

Pascal

```
function CancelEdit: Boolean; stdcall;
```

Description

Hide the node editor here. This might be something like `Visible := False` or `Hide`. The return value should be `True` if the editing can be cancelled. Return `false` if the node editor is in an internal state which does not allow to cancel the editing right now.

Do not destroy the node editor instance because this will be done implicitly via reference counting.

Notes

If the edited tree is changed during this function, i.e. focus change, node deletion and so on, `CancelEdit` might be

called again by the tree which can lead to access violations. It is therefore advisable to block reentrancy with a

boolean variable. Example:

```
function TStringEditLink.CancelEdit: Boolean;  
begin  
  Result := not FStopping;  
  if Result then  
  begin  
    FStopping := True;  
    FEdit.Hide;  
    FTree.CancelEditNode;  
  end;  
end;
```

Interface

[IVTEditLink Interface](#)

Links

[IVTEditLink Interface](#)

What do you think about this topic? [Send feedback!](#)



[IVTEditLink Interface](#) |
[IVTEditLink.CancelEdit Method](#) |
[IVTEditLink.GetBounds Method](#)

IVTEditLink.EndEdit Method

[IVTEditLink Interface](#)

This function will be called by the virtual tree when the current editing is being finished.

Pascal

```
function EndEdit: Boolean; stdcall;
```

Description

Hide the node editor here. This might be something like `Visible := False` or `Hide`. The return value should be true if the editing can be finished. Return false if the node editor is in an internal state which does not allow to finish the editing right now - possibly because there is no valid value available at the moment. If the editing can be finished transmit the edited value to the tree or to the data structure which is displayed in the tree.

Do not destroy the node editor instance because this will be done implicitly via reference counting.

Notes

If the edited tree is changed during this function, i.e. focus change, node deletion and so on, `EndEdit` might be called again by the tree which can lead to access violations. It is therefore advisable to block reentrancy with a boolean

variable. Example:

```
function TStringEditLink.EndEdit: Boolean;  
begin  
    Result := not FStopping;  
    if Result then  
        try  
            FStopping := True;  
            if FEdit.Modified then  
                FTree.DoNewText(FNode, FColumn, FEdit.Caption  
                FEdit.Hide;  
        except  
            FStopping := False;  
            raise;  
        end;  
end;
```

Interface

[IVTEditLink Interface](#)

Links

[IVTEditLink Interface](#)

What do you think about this topic? [Send feedback!](#)



[IVTEditLink Interface](#) |
[IVTEditLink.EndEdit Method](#) |
[IVTEditLink.PrepareEdit Method](#)

IVTEditLink.GetBounds Method

[IVTEditLink Interface](#)

The virtual tree can use this function to get the current bounding rect of the node editor.

Pascal

```
function GetBounds: TRect; stdcall;
```

Description

The bounding rect of the node editor may change during the editing to reflect the changed edit contents. The tree uses this function to query the current bounding rect of the editor. VCL components derived from TControl have a BoundsRect property which can be used as a return value here.

Interface

[IVTEditLink Interface](#)

Links

[IVTEditLink Interface](#)

What do you think about this topic? [Send feedback!](#)



[IVTEditLink Interface](#) |
[IVTEditLink.GetBounds Method](#) |
[IVTEditLink.ProcessMessage Method](#)

IVTEditLink.PrepareEdit Method

[IVTEditLink Interface](#)

This function is called by a virtual tree to initialize the node editor.

Pascal

```
function PrepareEdit(Tree: TBaseVirtualTree; Node: P
```

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 Interface](#) |
[IVTEditLink.PrepareEdit Method](#) |
[IVTEditLink.SetBounds Method](#)

IVTEditLink.ProcessMessage Method

[IVTEditLink Interface](#)

This function is used to forward messages being directed to the virtual tree.

Pascal

```
procedure ProcessMessage(var Message: TMessage); std
```

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

```
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:

```
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](#) | [Legend](#)

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.



Legend

Classes

Legend



Class



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

[TBaseVirtualTree Class](#) | [Legend](#)

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 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**

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.

  **ColumnsIsEmpty**

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 OnGetPopupMenu 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.
- 🟢🔗 **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 s 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.

  **HandleMouseDown**

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

[TBaseVirtualTree Class](#) | [Legend](#)

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.



Legend

TBaseVirtualTree Class

Legend



protected



Property



public



read only



Event



Method



virtual



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.OnColumnDbClick Event](#)

See Also

[OnColumnClick](#), [OnHeaderDbClick](#)



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.OnHeaderDbClick 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](#), [VTTTreeStreamVersion](#),
[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](#), [VTTTreeStreamVersion](#),
[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.ColumnsIsEmpty 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

[TBufferedString Class](#) | [Legend](#)

Methods

  **Add**

Not documented.

  **AddNewLine**

Not documented.

   **Destroy**

Not documented.



TBufferedString Class Properties

[TBufferedString Class](#) | [Legend](#)

Properties

 [AsString](#)

Not documented.



Legend

TBufferedString Class

Legend



public



Property



read only



Method



virtual



TClipboardFormatList Class Methods

[TClipboardFormatList Class](#) | [Legend](#)

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

TClipboardFormatList Class

Legend



public



Method



virtual



TClipboardFormats Class Methods

[TClipboardFormats Class](#) | [Legend](#)

Methods

 **Add**

Adds a new format to the internal list.

 **Create**

Constructor of the class.

 **Insert**

Adds a new format to the internal list.



TClipboardFormats Class Properties

[TClipboardFormats Class](#) | [Legend](#)

Properties

 **Owner**

Not documented.



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

[TCustomStringTreeOptions Class](#) | [Legend](#)

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.



TCustomStringTreeOptions Class Properties

[TCustomStringTreeOptions Class](#) | [Legend](#)

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

[TCustomVirtualDrawTree Class](#) | [Legend](#)

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.
- ➤ **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.
-



TCustomVirtualDrawTree Class Methods

[TCustomVirtualDrawTree Class](#) | [Legend](#)

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**
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.
-  **ColumnsIsEmpty**
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 OnGetPopupMenu 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.

  **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 s 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.

  **HandleMouseDbtClick**

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

[TCustomVirtualDrawTree Class](#) | [Legend](#)

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

[TCustomVirtualStringTree Class](#) | [Legend](#)

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.
- ➤ **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.
-



TCustomVirtualStringTree Class Methods

[TCustomVirtualStringTree Class](#) | [Legend](#)

Methods

 **AdjustPaintCellRect**

Method which can be used by descendants to adjust the given rectangle during a paint cycle.

 **CalculateTextWidth**

Not documented.

 **ColumnsEmpty**

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**
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.
-  **ColumnsEmpty**
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 OnGetPopupMenu 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.

  **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 s 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.



TCustomVirtualStringTree Class Properties

[TCustomVirtualStringTree Class](#) | [Legend](#)

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.



Legend

TCustomVirtualStringTree Class

Legend



protected



Property



read only



public



Event



Method



virtual



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.ColumnsIsEmpty 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

[TEnumFormatEtc Class](#) | [Legend](#)

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.
-



Legend

TScrollBarOptions Class

Legend



published



Property



public



Method



virtual



protected



TStringEditLink Class Methods

[TStringEditLink Class](#) | [Legend](#)

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

[TStringEditLink Class](#) | [Legend](#)

Properties

 [Edit](#)

Not documented.



Legend

TStringEditLink Class

Legend



public



Property



Method



virtual



TStringTreeOptions Class Methods

[TStringTreeOptions Class](#) | [Legend](#)

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

[TStringTreeOptions Class](#) | [Legend](#)

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.



Legend

TStringTreeOptions Class

Legend



published



Property



protected



public



read only



Method



virtual



TVirtualDrawTree Class Events

[TVirtualDrawTree 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.
- ➤ **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.

●> **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.

●> **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**
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.

➤ **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.



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 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**

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.
-  **ColumnsEmpty**
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 OnGetPopupMenu 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.

  **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 s 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.
- 🔗👤 **HandleMouseDbIClick**
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.
-



TVirtualDrawTree Class Properties

[TVirtualDrawTree Class](#) | [Legend](#)

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 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

TVirtualDrawTree Class

Legend

-  published
 -  Property
 -  public
 -  protected
 -  read only
 -  Event
 -  Method
 -  virtual
-



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.OnColumnDbClick Event](#)

See Also

OnColumnClick, OnHeaderDbClick



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.OnHeaderDbClick 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, VTTTreeStreamVersion,
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, VTTTreeStreamVersion,
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

[TVirtualStringTree 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.

●> **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.

●> **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.

➤ **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.

- ➤ **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.



TVirtualStringTree Class Methods

[TVirtualStringTree Class](#) | [Legend](#)

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.

ColumnsIsEmpty

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**

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.
-  **ColumnsEmpty**
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 OnGetPopupMenu 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.

  **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 s 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.



TVirtualStringTree Class Properties

[TVirtualStringTree Class](#) | [Legend](#)

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.

🌐 **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.

  **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.



Legend

TVirtualStringTree Class

Legend



published



Property



public



protected



read only



Event



Method



virtual



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

[TVirtualStringTree.NodeAlignment 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.OnColumnDblClick Event](#)

See Also

[OnColumnClick](#), [OnHeaderDblClick](#)



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.OnHeaderDbClick 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, VTTTreeStreamVersion,
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, VTTTreeStreamVersion,
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

[TVirtualTreeColumn Class](#) | [Legend](#)

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

[TVirtualTreeColumn Class](#) | [Legend](#)

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

[TVirtualTreeColumns Class](#) | [Legend](#)

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.

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 PositionToIndex array, which primarily determines where each column is placed visually.



TVirtualTreeColumns Class Properties

[TVirtualTreeColumns Class](#) | [Legend](#)

Properties

   **ClickIndex**
Not documented.

   **Header**
Not documented.

   **HeaderBitmap**
Not documented.

  **Items**
Not documented.

   **PositionToIndex**
Not documented.

   **TrackIndex**
Not documented.



Legend

TVirtualTreeColumns Class

Legend



public



Property



read only



protected



Method



virtual



TVirtualTreeHintWindow Class Methods

[TVirtualTreeHintWindow Class](#) | [Legend](#)

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

[TVirtualTreeOptions Class](#) | [Legend](#)

Methods

TCustomVirtualTreeOptions Class

  **AssignTo**

Used to copy this option class to another option collection.

  **Create**

Constructor of the class.



TVirtualTreeOptions Class Properties

[TVirtualTreeOptions Class](#) | [Legend](#)

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.



Legend

TVirtualTreeOptions Class

Legend



published



Property



protected



public



read only



Method



virtual



TVTColors Class Methods

[TVTColors Class](#) | [Legend](#)

Methods

 **Assign**

Not documented.

 **Create**

Not documented.



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.



Legend

TVTColors Class

Legend



published



Property



public



Method



virtual



TVTDataObject Class Methods

[TVTDataObject Class](#) | [Legend](#)

Methods

 **CanonicalUnknown**

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**

Searchs 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.



TVTDataObject Class Properties

[TVTDataObject Class](#) | [Legend](#)

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

[TVTDragImage Class](#) | [Legend](#)

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](#).



Add a summary here...



TVTDragImage Class Properties

[TVTDragImage Class](#) | [Legend](#)

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

[TVTDragManager Class](#) | [Legend](#)

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

  [AutoAdjustSize](#)

Not documented.

  [Create](#)

Not documented.

   [CreateParams](#)

Not documented.

   [Release](#)

Not documented.



TVTEdit Class Properties

[TVTEdit Class](#) | [Legend](#)

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.
-



Legend

TVTEdit Class

Legend



public



Property



protected



Method



virtual



TVTHeader Class Methods

[TVTHeader Class](#) | [Legend](#)

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 **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

[TVTHeader Class](#) | [Legend](#)

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.



Legend

TVTHHeader Class

Legend



published



Property



public



read only



Method



virtual



protected



TVTHeaderPopupMenu Class Events

[TVTHeaderPopupMenu Class](#) | [Legend](#)

Events

 [OnAddHeaderPopupMenuItem](#)

Not documented.

 [OnColumnChange](#)

Not documented.



TVTHeaderPopupMenu Class Methods

[TVTHeaderPopupMenu Class](#) | [Legend](#)

Methods

 [DoAddHeaderPopupMenuItem](#)

Not documented.

 [DoColumnChange](#)

Not documented.

 [OnMenuItemClick](#)

Not documented.

 [Popup](#)

Not documented.



TVTHeaderPopupMenu Class Properties

[TVTHeaderPopupMenu Class](#) | [Legend](#)

Properties

  **Options**

Not documented.



Legend

TVTHaderPopupMenu Class

Legend



published



Event



Property



protected



Method



virtual



public



TWideBufferedString Class Methods

[TWideBufferedString Class](#) | [Legend](#)

Methods

  **Add**

Not documented.

  **AddNewLine**

Not documented.

   **Destroy**

Not documented.



TWideBufferedString Class Properties

[TWideBufferedString Class](#) | [Legend](#)

Properties

 [AsString](#)

Not documented.



Legend

TWideBufferedString Class

Legend



public



Property



read only



Method



virtual



TWorkerThread Class Methods

[TWorkerThread Class](#) | [Legend](#)

Methods

  **AddTree**

Not documented.

  **ChangeTreeStates**

Not documented.

  **Create**

Not documented.

   **Destroy**

Not documented.

   **Execute**

Not documented.

  **RemoveTree**

Not documented.



TWorkerThread Class Properties

[TWorkerThread Class](#) | [Legend](#)

Properties

 [CurrentTree](#)
Not documented.



Legend

TWorkerThread Class

Legend



public



Property



read only



Method



protected



virtual



Functions

[Functions](#) | [Legend](#)

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.



Legend

Functions

Legend

Function



Structs and Records Enumerations

[Structs and Records](#) | [Legend](#)

Enumerations

 [TVTTooltipLineBreakStyle](#)

Not documented.



Structs and Records Records

[Structs and Records](#) | [Legend](#)

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

[Types](#) | [Legend](#)

Enumerations

❖ TAddPopuItem`Type`

Not documented.

❖ TBlend`Mode`

Not documented.

❖ TChange`Reason`

Not documented.

❖ TCheckImage`Kind`

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

❖ TCheck`State`

Returns the current state of a node's check box, radio button or node button.

❖ TCheck`Type`

Not documented.

❖ TDrag`Operation`

Not documented.

❖ TDrop`Mode`

Not documented.

❖ THeader`State`

Not documented.

❖ THintAnimation`Type`

Not documented.

❖ THit`Position`

Not documented.

❖ TItemErase`Action`

Not documented.

❖ TScrollBar`Style`

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

[Types](#) | [Legend](#)

Types

- ◆ PCardinal
Not documented.
- ◆ PClipboardFormatListEntry
Not documented.
- ◆ PSHDragImage
Not documented.
- ◆ PVirtualNode
Not documented.
- ◆ PVTHintData
Not documented.
- ◆ PVTReference
Not documented.
- ◆ TAddHeaderPopuItemEvent
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.
- ◆ TVTColumnDbClickEvent
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.
 - ◆ TVTGetCellsIsEmptyEvent
Not documented.
 - ◆ TVTGetImageExEvent
Not documented.
 - ◆ TVTMenuItem
Not documented.
 - ◆ TVTScrollbarShowEvent
Not documented.
-



Legend

Types

Legend



Type



Struct



Variables

[Variables](#) | [Legend](#)

Variables

- ◆ **CF_CSV**
Not documented.
- ◆ **CF_HTML**
Not documented.
- ◆ **CF_VIRTUALTREE**
Not documented.
- ◆ **CF_VRTF**
Not documented.
- ◆ **CF_VRTFNNOOBS**
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.

◆ NeedToUnitalize

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

Variables

Legend



Variable



Constants

[Constants](#) | [Legend](#)

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.

◆ **VTHeaderStreamVersion**

Not documented.

◆ **VTTreeStreamVersion**

Not documented.

◆ **VTVersion**

Not documented.

◆ **WideCR**

Not documented.

◆ **WideLF**

Not documented.

◆ **WideLineSeparator**

Not documented.

◆ **WideNull**

Not documented.

◆ **WM_CHANGESTATE**

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](#)



Legend

Constants

Legend



Constant



Symbol Reference Interfaces

[Symbol Reference](#) | [Legend](#)

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

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.



IVTDragManager Interface Properties

[IVTDragManager Interface](#) | [Legend](#)

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

[IVTEditLink Interface](#) | [Legend](#)

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

Legend



public



Method



Legend



Class



Legend



protected



Event



Legend



public



Method



protected



virtual



Legend



protected



Property



public



read only



Legend



public



Method



virtual



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



public



Method



virtual



protected



Legend



published



Property



public



read only



Legend



public



Method



virtual



protected



Legend



public



Property



read only



protected



Legend



public



Method



virtual



protected



Legend



public



Method



virtual



Legend



published



Property



protected



public



read only



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

Function



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



Legend



public



Method
