

TMS Memo for FireMonkey
DEVELOPERS GUIDE

June 2015
Copyright © 2015 by tmssoftware.com bvba
Web: http://www.tmssoftware.com
Email: info@tmssoftware.com

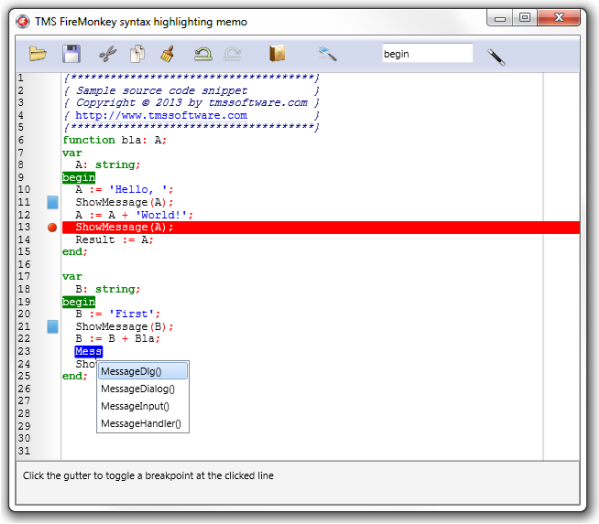
Index

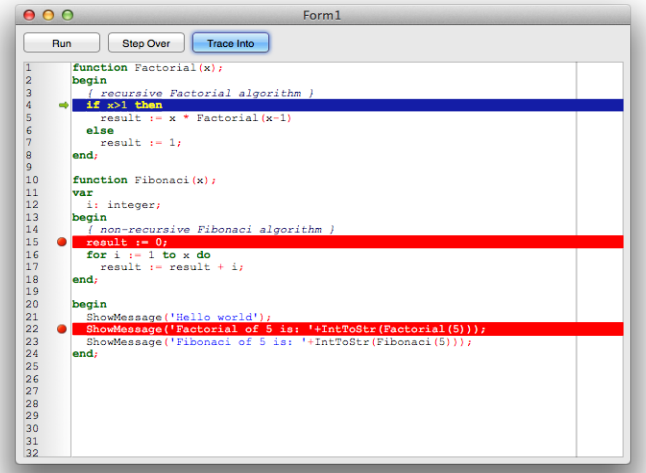
Introduction 4
List of included components 5
TTMSFMXMemo 6
TTMSFMXMemo description 6
TTMSFMXMemo features 6
TTMSFMXMemo architecture 7
TTMSFMXMemo use 8
TTMSFMXMemo published properties 8
TTMSFMXMemo public properties 9
TTMSFMXMemo methods 9
TTMSFMXMemo events 11
TTMSFMXMemo style template 12
TTMSFMXMemo style template elements 12
TTMSFMXActiveLineSettings 13
TTMSFMXActiveLineSettings description 13
TTMSFMXActiveLineSettings properties 13
TAutoCompletion 13
TAutoCompletion description 13
TAutoCompletion properties 14
TTMSFMXGutter 14
TTMSFMXGutter description 14
TTMSFMXGutter properties 15
TTMSFMXMemo.Lines 15
TTMSFMXMemoFindDialog 16
TTMSFMXMemoFindDialog description 16
TTMSFMXMemoFindDialog properties 16
TTMSFMXMemoFindDialog Methods 16
TTMSFMXMemoFindDialog Events 16
TTMSFMXMemoFindAndReplaceDialog 17
TTMSFMXMemoFindAndReplaceDialog description 17
TTMSFMXMemoFindAndReplaceDialog properties 17
TTMSFMXMemoFindAndReplaceDialog Events 17
Syntax Stylers 18
TTMSFMXMemoPascalStyler 18
TTMSFMXMemoBasicStyler 22
TTMSFMXMemoCSharpStyler 22
TTMSFMXMemoCSSStyler 22
TTMSFMXMemoHTMLStyler 22
TTMSFMXMemoJavaScriptStyler 22
TTMSFMXMemoSQLMemoStyler 22
TTMSFMXMemoWebMemoStyler 22

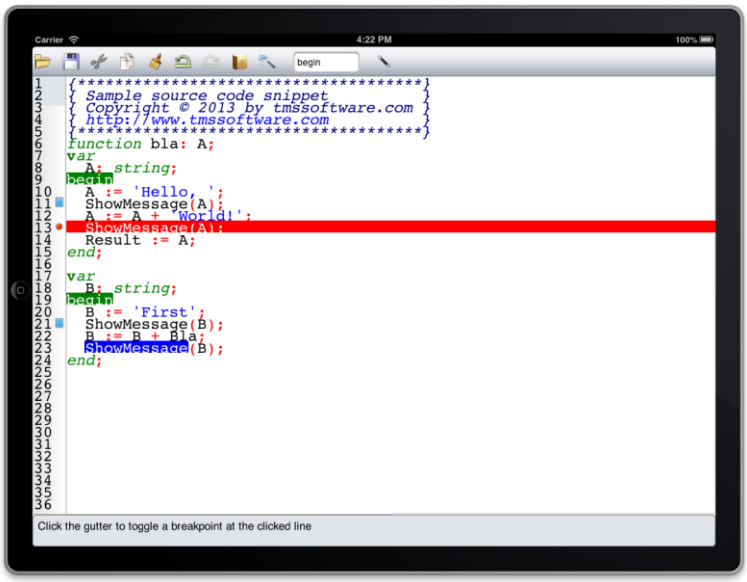
TTMSFMXMemoXMLMemoStyler 22

Introduction

The TMS Memo for FireMonkey offers a fully cross-platform, versatile and feature-packed syntax highlighting memo for the Embarcadero cross-platform framework FireMonkey. As such, it is designed for use with Win32, Win64, Mac OS-X, iOS and Android operating systems. It is designed to respect the philosophy of style-able controls. At the same time, it is sufficiently similar to the VCL TAdvMemo to make developers used to TAdvMemo quickly familiar and up & running.







IMPORTANT NOTICE:
If the FireMonkey framework is new to you, please see the chapter “General FireMonkey component usage guidelines” that offers an introduction that is recommended to read before you start working with the TMS Memo for FireMonkey. Another interesting source of information is http://docwiki.embarcadero.com/RADStudio/en/FireMonkey_Application_Platform

List of included components

TTMSFMXMemo: Is the core syntax highlighting memo component and offers syntax highlighting for a large number of programming languages including the customizability of syntax highlighting for custom scripting languages.

TTMSFMXMemoFindDialog: Allows searching for a certain (sub-) string in the TTMSFMXMemo control.

TTMSFMXMemoFindAndReplaceDialog: Allows searching for a certain (sub-) string in the TTMSFMXMemo control, and replace it with another text string.

TAdvMemoCapitalChecker: Is an extension of the standard TAdvMemo that allows checking for capital letters at the start of a sentence. This feature is useful when creating a standard text editor.

TTMSFMXMemoBasicStyler: Component holding syntax settings for the MS Visual Basic language.

TTMSFMXMemoCSharpStyler: Component holding syntax settings for the C# files syntax checking.

TTMSFMXMemoCSSStyler: Component holding syntax settings for CSS files.

TTMSFMXMemoHTMLStyler: Component holding syntax settings for HTML files.

TTMSFMXMemoJavaScriptStyler: Component holding syntax settings for the JavaScript language.

TTMSFMXMemoPascalStyler: Component holding syntax settings for the Pascal language.

TTMSFMXMemoSQLStyler: Component holding syntax settings for SQL files.

TTMSFMXMemoWebStyler: Component holding syntax settings for Web files (HTML, CSS, Javascript).

TMSFMXMemoXMLStyler: Component holding syntax settings for XML files.

TTMSFMXMemo

TTMSFMXMemo description

The TTMSFMXMemo is a highly configurable syntax highlighting memo control.

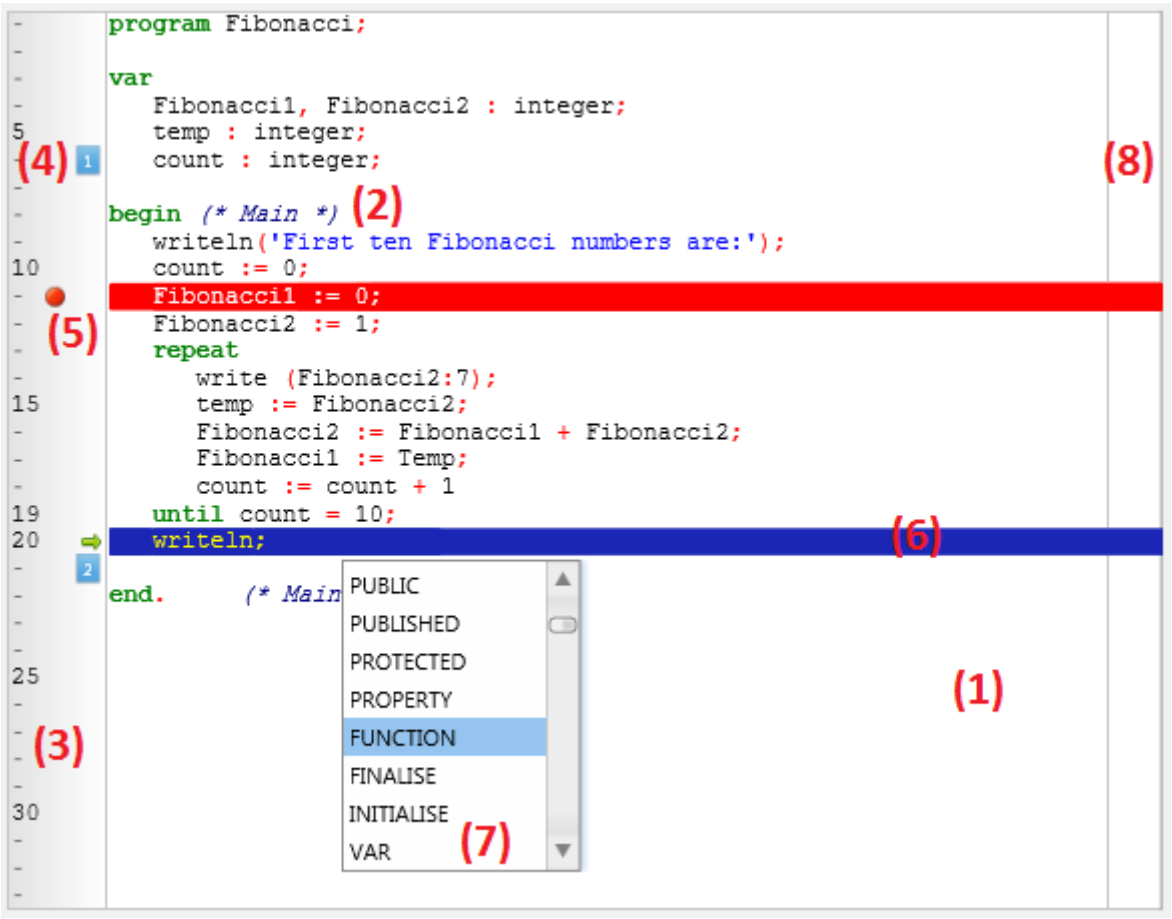
The TTMSFMXMemo has features like auto completion and auto correction, bookmarks support, markers and breakpoint indication and can optionally interface to a source code container for easy & memory friendly multi-source editing.

The TMS TTMSFMXMemo component comes with built-in support for a wide range of programming languages and custom language syntax highlighting can be added.

TTMSFMXMemo features

- Lightweight memo control with configurable syntax highlighting
- Highlighting for Basic, C#, CSS, HTML, JavaScript, PascalSQL, Web and XML files.
- Undo and redo functions
- Optional gutter with configurable line number display
- Clipboard operations
- Find and replace dialogs
- Save to formatted HTML support
- Configurable auto-completion
- URL aware
- Search highlight, search with expressions

TTMSFMXMemo architecture



The core part of the TTMSFMXMemo control is the editor with (1) syntax highlighting (2). A gutter can be displayed that shows line numbers (3) and additional information such as a bookmark (4) and a breakpoint (5).
The active line (6) can be indicated with an active line indicator (green arrow on the left hand side) and/or a highlighted blue line. Code completion (7) can be activated by pressing the Ctrl + Spacebar keys or can automatically appear for class methods, events, properties. An optional right margin indicator is drawn (8).

TTMSFMXMemo use

Getting started

From the component palette, select the TTMSFMXMemo control and drop it on a form.
This shows an empty memo, with a gutter on the left hand side.

With the default settings, TTMSFMXMemo behaves similar to the standard FireMonkey TMemo control. The content of the editor is handled by the Lines: TStringList property. Content of the memo can be get and set via this property. The caret position in the memo can be get and set by the properties TMSFMXMemo.CurX: integer, TMSFMXMemo.CurY: integer. The selection is get and set with TMSFMXMemo.SelStart: integer and TMSFMXMemo.SelLength: integer properties and the selected text with the property TMSFMXMemo.Selection: string. The selection can also be retrieved in X, Y coordinates with TMSFMXMemo.SelStartX, TMSFMXMemo.SelStartY, TMSFMXMemo.SelEndX, TMSFMXMemo.SelEndY.

TTMSFMXMemo published properties

- ActiveLineSettings: Class property of TAdvActiveLineSettings type. (See detailed information in the TActiveLineSettings paragraph).
- AutoCompletion: Class property of TAutoCompletion type. (See detailed information in the TAutoCompletion paragraph).
- AutoCorrect: Class property of TAutoCorrect type (See detailed information in the TAutoCorrect paragraph).
- AutoExpand: Boolean: When set to true, when clicked after the end of the line, the line will be filled with spaces till the X position clicked. When false, the cursor is placed directly after the last character of a line, irrespective of the X position clicked after a line.
- AutoIndent: Boolean: When set to true, pressing enter to start a new line will automatically start the new line at the same indent as the previous line.
- CaseSensitive: Boolean: When set to true, the TTMSFMXMemo control is case sensitive with respect to keyword highlighting.
- CharCase: TCharCase: Selects the default case of text entered in the memo (similar to a TEdit or TMemo)
- DelErase: Boolean: When set to true, the selected area is erased when typing, when set to false only the first character where typed is erased.
- EnhancedHomeKey: Boolean: When set to true, the cursor goes to the start / end of the line without taking the spaces before or after the line in account.
- Gutter: Class property of TTMSFMXGutter type. (More detailed information can be found in the TTMSFMXGutter paragraph).
- HideSelection: Boolean: When set to true, the selection is hidden when the focus leaves the memo control.
- Lines: Class property of TTMSFMXMemoStrings type. (More detailed information can be found in the TTMSFMXMemoStrings paragraph).
- LineSpacing: integer: Sets the spacing (in pixels) between lines in the memo. Default value is 2.
- RightMargin: Integer: Defines the position of the right margin.
- ShowBands: Boolean: When true, the odd lines in the memo are shown in a different color (band color)
- ShowRightMargin: Boolean: When set to true, a right vertical margin is shown.
- SmartTabs: Boolean: When set to true, the tabs are automatically set to a multiple of the TTMSFMXMemo.TabSize. When set to false, the TTMSFMXMemo.TabSize is added from the current position of the cursor.

- SyntaxStyles: TTMSFMXCustomMemoStyler: Sets the styler to one of the included syntax stylers components for automatic syntax based highlighting. One can choose out of the following included syntax stylers: TTMSFMXMemoBasicStyler, TTMSFMXMemoCSharpStyler, TTMSFMXMemoCSSStyler, TTMSFMXMemoHTMLStyler, TTMSFMXMemoJavascriptStyler, TTMSFMXMemoPascalStyler, TTMSFMXMemoSQLStyler, TTMSFMXMemoWebStyler, TTMSFMXMemoXMLStyler. Alternatively, a custom syntax styler can be built by descending from TTMSFMXCustomMemoStyler.
- TabSize: Integer: Sets the number of tabulation characters when pressing the Tab button.
- TrimTrailingSpaces: Boolean: When set to true, trailing spaces are removed.
- UndoLimit: Integer: Holds the maximum number of Undo steps.
- UndoLineByLine: Boolean: When set to true, an Undo is performed one line at a time.
- UrlAware: Boolean: When set to true, an URL is recognized by the control, and displayed in a specific style.
- UseStyler: Boolean: When set to true, the assigned syntax styler settings will be applied to the TTMSFMXMemo control.
- Version: String: Returns the version of the component as string.
- WantTab: Boolean: When set to true, instead of jumping to the next control, tabulation functionality is attributed to the TTMSFMXMemo control.

TTMSFMXMemo public properties

- CurX: Integer: Gets / sets the caret column position.
- CurY: Integer: Gets / sets the caret row position.
- Selection: String: Gets / sets the selected text.
- SelStart: Integer: Gets / sets the start position of the selected text.
- SelLength: Integer: Gets / sets the length of the selected text
- SelStartX: Integer: Gets / sets the starting column of the selected text.
- SelStartY: Integer: Gets / sets the starting row of the selected text.
- SelEndX: Integer: Gets / sets the ending column of the selected text.
- SelEndY: Integer: Gets / sets the ending row of the selected text.
- ActiveLine: Integer Returns the active line number.
- Bookmark[LineIndex: Integer]: Boolean: When set to true, the line at LineIndex position holds a bookmark.
- Bookmarks[Index: Integer]: Integer: Holds the line index for that bookmark number Index.
- BookmarkIndex[LineIndex: integer]: Integer: Holds the index number for the bookmark at LineIndex.
- BreakPoint[Index: Integer]: Boolean : When set to true, the line at index position holds a breakpoint.
- OverWrite: Boolean: When set to true, the entered text overwrites existing one, if set to false, text is inserted at the cursor position.
- TopLine: Integer: Gets / sets the first visible text-line on the top row of the editor.
- LeftCol: Integer: Gets / sets the first visible text-column on the most left position of the editor.
- UndoList: TAdvUndoList: Class property of TList type, holding all actions for undo in the memo.

TTMSFMXMemo methods

- Procedure MouseToCursor(X, Y: Integer; var CursorX, CursorY: Integer): Returns the text-column CursorX and text-line CursorY position for the given X, Y mouse position in the editor.
- Procedure CopyToClipboard: Copies the selected text to the clipboard.
- Procedure PasteFromClipboard: Inserts the text from the clipboard to the TAdvMemo control.
- Procedure CutToClipboard: Cuts the selected text to the clipboard.
- Function IsEmpty: Boolean: Returns True if the TTMSFMXMemo.Editor is empty.

- Procedure SelectAll: Selects al the text in the TTMSFMXMemo editor.
- Procedure ClearBreakPoints: Clears the breakpoint list, resulting in the removing of all breakpoints.
- Function CharFromPos(X, Y: Integer): TFullPos: Returns the exact position of the character at the actual mouse cursor position.
- Procedure PosFromText(TextPos: Integer, var X, Y: integer): Converts the TextPos text position in the memo to X,Y cursor coordinates.
- Procedure TextFromPos(X, Y: Integer; var TextPos: Integer): Converts the cursor coordinates X,Y to the text position in the memo.
- Procedure DeleteSelection: Deletes the current selection/selected text in the editor.
- Procedure BlockIndent(Fromline, ToLine, Indent: Integer; AllowUndo: Boolean = True): Sets indentation for a block of source-code, starting at FromLine, and extending to ToLine. Ident sets the width of the indentation in characters. When AllowUndo is set to True, the indentation can be reversed.
- Function WordAtXY(X, Y: Integer): string: Returns the word at the given X,Y cursor coordinates in the memo.
- Function TokenAtXY(X, Y: Integer): string: Returns the token at the given X,Y cursor coordinates in the memo.
- Function FullWordAtCursor: string: Returns the full word at the given X,Y cursor coordinates in the memo.
- Procedure ClearSelection: Clears the current selection in the memo.
- Procedure ClearBookmarks: Clears the bookmark list, removing all previously added bookmarks.
- Procedure ClearUndoRedo: Clears the undo-redo list.
- Procedure GotoBookmark(Index: Integer): Positions the cursor at the chosen bookmark.
- Function HasBookmarks: Boolean: Returns true if bookmarks have been created for the TAdvMemo.
- Function FindTextCount(SearchStr: String, Options: TTMSFMXFindDialogOptions): Integer: Returns the number of occurrences of SearchStr, depending on the parameters set in TTMSFMXFindDialogOption:
- TTMSFMXFindDialogOption = (fdoCaseSensitive, fdoExpression, fdoCloseIfFound,
- fdoWholeWordOnly, fdoWrapAtEndOfFile, fdoContinueToNextFile, fdoMoreEnabled,
- fdoSetMarkerEnabled, fdoPreviousEnabled, fdoFindEnabled, fdoCloseEnabled, fdoMoreExpanded, fdoFindList, fdoDown, fdoSelection);

- Function FindText(SearchStr: String, Options: TTMSFMXFindDialogOptions): Integer: Returns the position of searchstr in the memo, depending on the parameters set in TTMSFMXFindDialogOptions, as described in FindTextCount.
- Function FindAndReplace (SearchStr, NewStr: String, Options: TTMSFMXFindDialogOptions): Integer: Searches for SearchStr in the editor, depending on the parameters set in TTMSFMXFindDialogOptions, and replaces the SearchStr with NewStr text.
- Procedure Clear: Clears all text from the TTMSFMXMemo editor.
- Procedure Undo: Reverses the last performed action in the memo. The number of undo-entries is limited with the TTMSFMXMemo.UndoLimit property;
- Procedure Redo: Reapplies the last reversed action.
- Function CanUndo: Boolean; Returns if the undo feature is activated in the TTMSFMXMemo editor screen.
- Function CanRedo: Boolean; Returns if the redo feature is activated in the TTMSFMXMemo editor screen.
- Function CanCut: Boolean; Returns if the cut feature is activated in the TTMSFMXMemo editor screen.
- Function CanPaste: Boolean; Returns if the paste feature is activated in the TTMSFMXMemo editor screen.
- Function SaveToHTML(Filename: String; Fixedfonts: Boolean = True): Boolean: Saves the text from the editor to a HTML formatted file with name Filename. Fixed fonts are used by default.

- Function SaveToHTMLStream(AStream: TMemoryStream; Fixedfonts: Boolean = True): Boolean: Saves the text from the editor to a HTML formatted memory stream. Fixed fonts are used by default.
- Function SaveToRTF(Filename: String; Fixedfonts: Boolean = True): Boolean: Saves the text from the editor to a Rich Text Format formatted file with name Filename. Fixed fonts are used by default.
- Function SaveToRTFStream(AStream: TMemoryStream; Fixedfonts: Boolean = True): Boolean: Saves the text from the editor to a Rich Text Format formatted memory stream. Fixed fonts are used by default.
- Procedure BeginUpdate; Freezes the content of the TAdvMemo, enabling to perform a large number of modifications to the text, without the redrawing of every modification, until the EndUpdate procedure is called. This enhances the performance.
- Procedure EndUpdate;
- Function GetVersionNr: Integer: Returns the version number.
- Function GetVersionString: String: Returns the version in a string format.

TTMSFMXMemo events

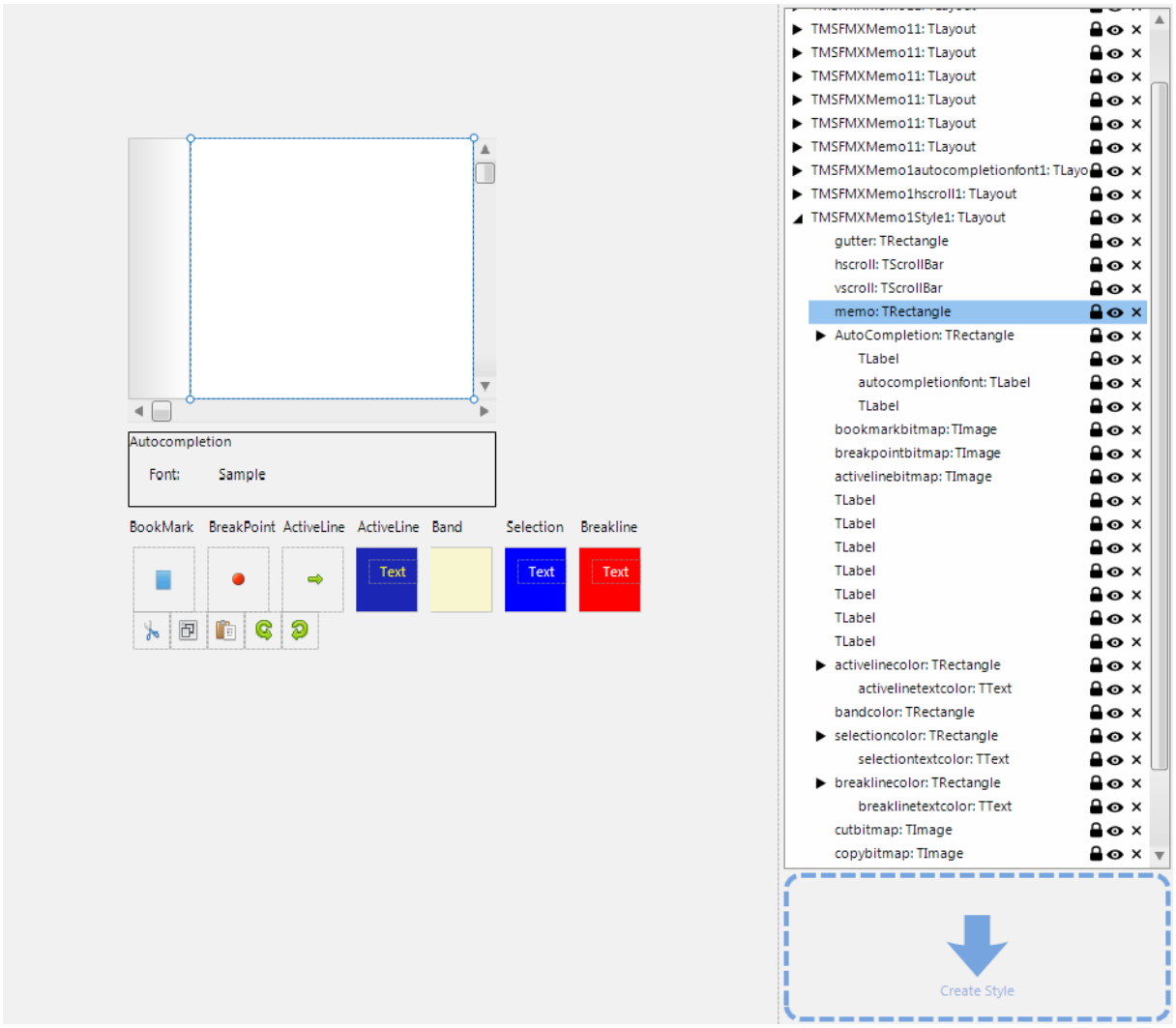
This is the list of events that the TTMSFMXMemo adds to the standard events already available in base class FireMonkey visual controls.

- OnAnchorClick: Is triggered a hyperlink in the memo text is clicked.
- OnAutoCompletion: Is triggered when AutoCompletion is started.
- OnAutoCompletionCustomizeItem : Event triggered for every item in the auto completion dropdown to allow for dynamic customization
- OnBeforeAutoCompletion: Is triggered before AutoCompletion is started.
- OnCancelAutoCompletion: Is triggered when AutoCompletion is canceled.
- OnClick: Is triggered on mouse click on the memo
- OnClipboardAction: Is triggered on Ctrl + C is pressed, and selected text is moved to the clipboard.
- OnContextMenuItemClick: Is triggered when an item from the context menu is selected.
- OnCursorChange: Is triggered whenever the cursor position changes.
- OnDblClick: Is triggered when the memo is double-clicked.
- OnGetAutoCompletionList: Is triggered when the auto completion list is loaded and allows for dynamic customization of this list.
- OnGutterClick: Is triggered when a click occurred in the gutter area.
- OnGutterDblClick: Is triggered when a double click occured in the gutter area.
- OnInsertAutoCompletionEntry: Is triggered when an auto completion entry is inserted.
- dynamically alter the line background.
- OnOverWriteToggle: Is triggered when the insert/overwrite is toggled.
- OnReplace: Is triggered when a text is replaced through a find & replace action.
- OnSelectionChange: Is triggered when the selection in the memo has changed.
- OnShowContextMenu: Is triggered just before the context menu is being shown and allows for application level customization of the context menu.
- OnUndoChange: Is triggered when changes in the text are undone.

TTMSFMXMemo style template

TTMSFMXMemo style template elements

The TTMSFMXMemo is designed according to the FireMonkey philosophy. As such, the template that make up the visual appearance of the component can be edited in the IDE via the style editor. The memo itself is constructed from a TLayout that holds a gutter rectangle, a memo rectangle, horizontal and vertical scrollbars. In addition, many different style elements are added that control the visual appearance of the different elements in the memo such as the selection color, breakpoint indicator, active line color, etc…



As such, changing the background color of the memo is done by opening the style editor, selecting the memo TRectangle style element and changing its color via memo.Fill.Color.
Note that this style element is also accessible at runtime via code. This style template element is retrieved with:
memo.GetMemoRect: TRectangle

and as such, its color can be changed in code with:

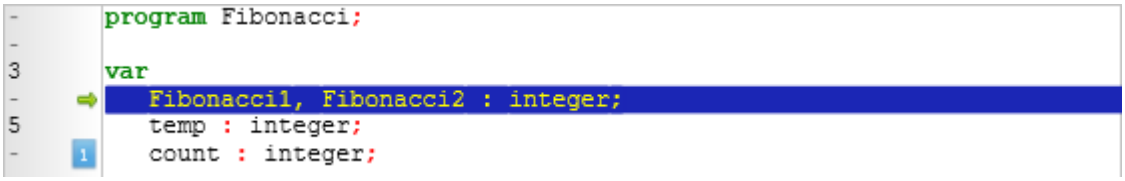
```
TMSFMXMemo1.GetMemoRect.Fill.Color := TAlphaColorRec.Aqua;
```

Further, it can be seen in the style editor that the font can be configured for the items in the autocompletion dropdown, the icons for markers, breakpoints and active line in the gutter. The background color and the text color of active lines, selection, breakpoint lines and optional color banding can be configured as well. Finally, the icons that are used in the context menu are also exposed in the style editor.

TTMSFMXActiveLineSettings

TTMSFMXActiveLineSettings description

The TTMSFMXActiveLineSettings class property controls how highlighting of the line position of the cursor is done. This highlighting can be done by a colored background and font, or by displaying an arrow in the gutter on the left. For maximum visibility, both possibilities can be set at the same time.



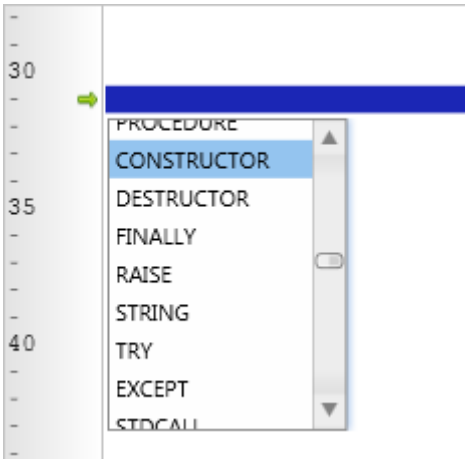
TTMSFMXActiveLineSettings properties

- ActiveLineAtCursor: When set to true, the active line at the cursor position is highlighted.
- ShowActiveLine: When set to true, displays the active line in color, as defined with ActiveLineColor and ActiveLineTextColor.
- ShowActiveLineIndicator: When set to true, shows a green arrow next to the line number in the gutter.

TAutoCompletion

TAutoCompletion description

The TAutoCompletion collection property allows to control the automatic completion of the edited text, depending on the chosen TTMSFMXMemo.SyntaxStyles component. Auto completion can be triggered with the keyboard by pressing Ctrl-Space. It can also be shown when the class separator character is pressed, configured by default to '.'



TAutoCompletion properties

- Active: When set to true, the auto completion feature is activated.
- AutoCompletionCount: Sets the maximum height of the autocompletion dropdown in number of items displayed.
- AutoWidth: When set to true, the width of the auto completion listbox is adapted according to the width of the items it holds.
- FromFirstChar: When set to true, auto completion can start immediately from the first entered character.
- Height: Sets the height of the auto completion listbox.
- MaxWidth: Sets the maximum with of the auto completion listbox. When the value is 0, this is ignored and the width of the dropdown is automatically adapted to fit all items.
- SizeDropDown: When set to true, the dropdown of the auto completion listbox can be sized by dragging the bottom right corner.
- StartToken: Holds the start characters that trigger the display of the auto completion listbox
- Width: Sets the width of the auto completion listbox.

TTMSFMXGutter

TTMSFMXGutter description

The TTMSFMXGutter is a collection of properties defining the layout of the gutter, a vertical pane on the left side of the memo control.



TTMSFMXGutter properties

- DigitCount: Sets the number of leading zeros used in line numbers.
- Font: Sets the default font that will be used for the gutter.
- LineNumberAt: Default this value is 1, meaning that the line number is shown on every line. When LineNumberAt would be set to 10, this would cause the line number to be displayed only every 10th line.
- LineNumberStart: Sets the start value of the line number. This allows to override the default line numbering that starts from line 1.
- NumberSuffix: Defines a suffix for the numbers.
- ShowLeadingZeros: When set to true, line numbers are displayed with leading zeroes, as many as defined in the DigitCount.
- ShowLineNumbers: When set to true, line numbers are displayed in the gutter.

TTMSFMXMemo.Lines

Adding or removing lines can be done either programmatically or at design-time. Double-click the TTMSFMXMemo.Lines property and the TTMSFMXMemo editor will popup, allowing adding new or removing existing lines.

A text can be copy-pasted to the TTMSFMXMemo Editor.

Adding/Removing lines programmatically

Adding a line:

```
TMSFMXMemo1.Lines.Add('Manually added line');
```

Removing the active line:

```
TMSFMXMemo1.Lines.Delete(AdvMemo1.ActiveLine);
```

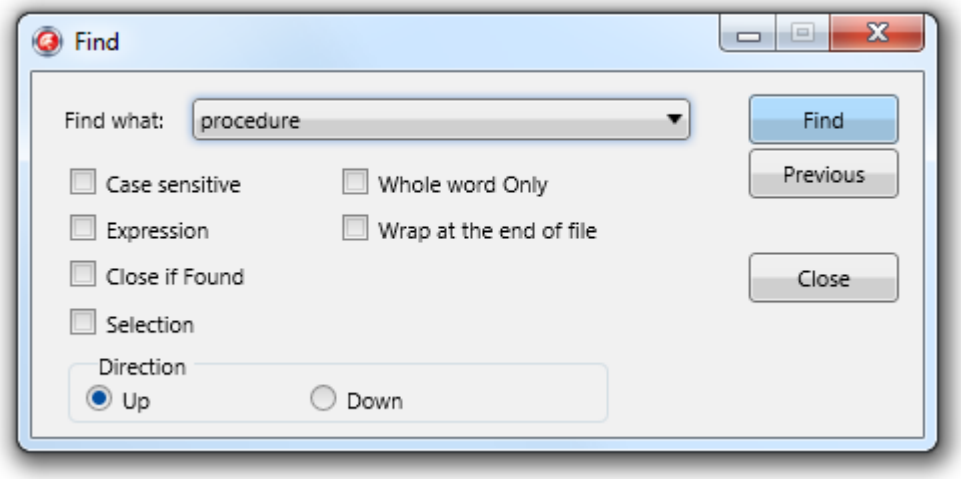
Loading a file:

```
  if OpenDialog1.Execute then
    TMSFMXMemo1.Lines.LoadFromFile(OpenDialog1.FileName);
```

TTMSFMXMemoFindDialog

TTMSFMXMemoFindDialog description

The TTMSFMXMemoFindDialog is a control that gives the possibility to search for a (sub)-string in the TTMSFMXMemo.



TTMSFMXMemoFindDialog properties

- Memo: The TTMSFMXMemo control where the string will be searched for.
- AutoHighlight: When true, the matching occurences of strings will be automatically highlighted while typing in the TTMSFMXFindDialog.
- DisplayMessage: When set to true, displays messages in the dialog box.
- FindText: Holds the substring to find in the text.
- FocusMemo: When set to true, the focus is automatically given back to the memo when a string is found.
- NotFoundMessage: Holds the text that is displayed when the substring is not found.
- Options: Sets the various options to control the find action.

TTMSFMXMemoFindDialog Methods

- Procedure Execute: Starts the search in the TTMSFMXMemo control for the (sub-) string defined in TTMSFMXMemo.FindText.
- Procedure CloseDialog: Close the CloseDialog programmatically.

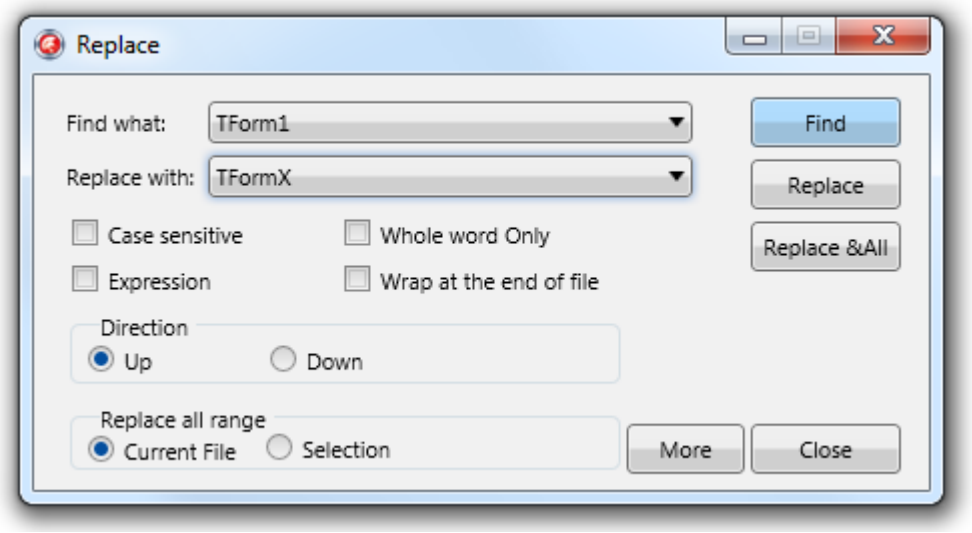
TTMSFMXMemoFindDialog Events

- OnFindDone: Is triggered when the search is finished.
- OnFindText: Is triggered when the text is found.
- OnShow: Is triggered when the dialog is first shown.
- OnClose: Is triggered when the dialog is closed.

TTMSFMXMemoFindAndReplaceDialog

TTMSFMXMemoFindAndReplaceDialog description

The TTMSFMXMemoFindAndReplaceDialog gives the possibility to search for a certain string in the TTMSFMXMemo editor and to replace occurences of the string with another one.



TTMSFMXMemoFindAndReplaceDialog properties

- Memo: The TTMSFMXMemo control where the string will be searched for.
- AutoHighlight: When true, the matching occurences of strings will be automatically highlighted while typing in the TTMSFMXReplaceDialog.
- DisplayMessage: When set to true, displays messages in the dialog box.
- FindText: Holds the substring to find in the text.
- FocusMemo: When set to true, the focus is given to the memo.
- NotFoundMessage: Holds the text that is displayed when the substring is not found.
- ReplaceDialog: Optionally can be set to an instance of TAdvReplaceDialog that offers extra functionality compared to the regular VCL TReplaceDialog.
- ReplaceText: Holds the text used for replacing the FindText substring.

TTMSFMXMemoFindAndReplaceDialog Events

- OnFindDone: Is triggered when the search is finished.
- OnFindText: Is triggered when the text is found.
- OnShow: Is triggered when the dialog is first shown.
- OnClose: Is triggered when the dialog is closed.

Syntax Stylers

TTMSFMXMemoPascalStyler

Description

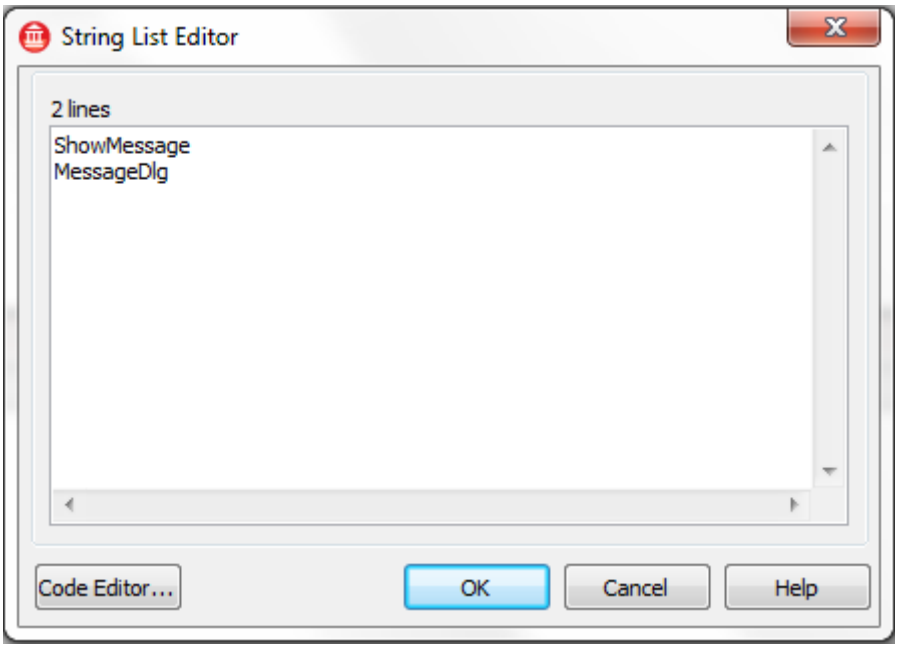
The TTMSFMXMemoPascalStyler is the syntax styler for Pascal files.

~~General~~ **TTMSFMXMemoPascalStyler** properties
~~settings~~

- ~~TAdvPascalMemoStyler properties~~

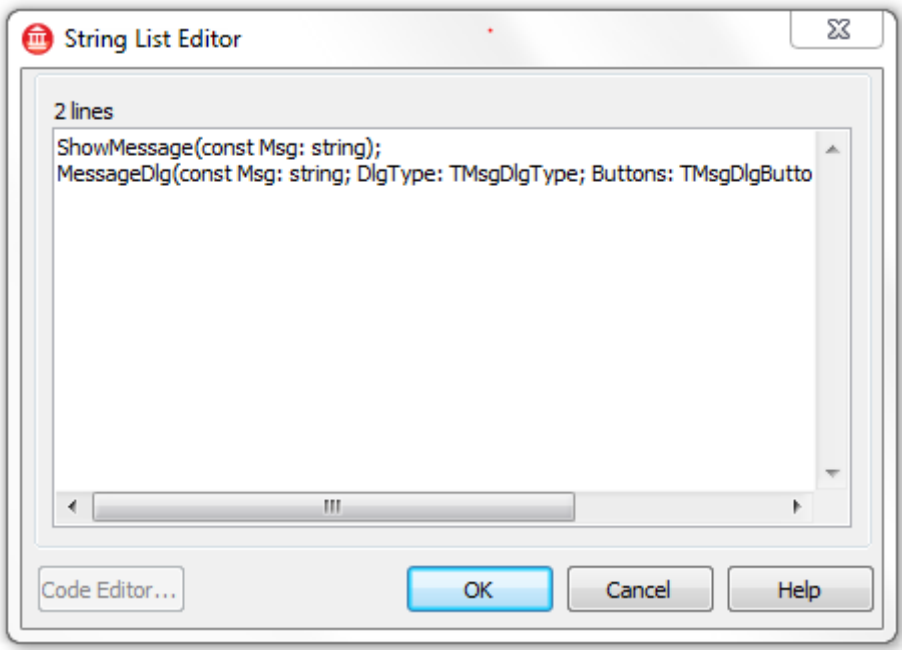
~~—~~

- **AllStyles:** Is a collection of TElementStyles (detailed information can be found in the TElementStyles paragraph).
- **AutoCompletion:** Is a TStringlist holding autocompletion text. When double clicking on the property, a string list editor is loaded, where autocompletionstrings can be added or removed.



- **BlockEnd:** Holds the text that defines the end of a code block.
- **BlockStart** Holds the text(s, comma separated) that are recognized as the start of a code block.
- **CommentStyle:** Is of type TCharStyle, consisting of:
  - **BKColor:** Sets the background color
  - **Style:** Style has four properties that can be enabled or disabled: fsBold, fsItalic, fsUnderline and fsStrikeOut.
  - **TextColor:** Sets the font color

- **DefaultExtension:** Holds the default extension for a Pascal ~~File~~file.
- **Description:** Holds a default description for the TTMSFMXMemoStyler.
- **Extensions:** Holds the possible (point-comma delimited) valid Pascal file extensions.
- **Filter:** Holds a filter for use with an openfiledialog, containing the valid ~~pascal~~ Pascal file extensions.
- **HexIdentifier:** Holds the character that identifies a hex value.
- **HintParameter:** Is of type THintParameter, consisting of:
    - **BKColor:** Sets the background color.
    - **HintCharDelimiter:** Holds the hint character delimiter.
    - **HintCharEnd:** Holds the hint end character.
    - **HintCharStart:** Holds the hint start character.
    - **HintCharWriteDelimiter:** Holds the hint character write delimiter.
    - **Parameters:** Is a TStringlist, holding the values of the hint parameters.

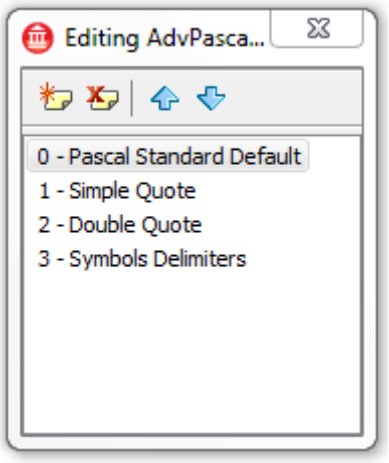


    - **TextColor:** Sets the font color for the hint.
    - **LineComment:** Holds the characters that identify the single line comment.
    - **MultiCommentLeft:** Holds the start character(s) that identifies the start of a multiline comment.
    - **MultiCommentRight:** Holds the end character(s) that identify the end of a multiline comment.
    - **Name:** Holds the name of the control.

- **NumberStyle:** Is of type TCharStyle, consisting of:
  - **BKColor:** Sets the background color
  - **Style:** Style has four properties that can be enabled or disabled: fsBold, fsItalic, fsUnderline and fsStrikeOut.
  - **TextColor:** Sets the font color.
- **RegionDefinitions:** Is a collection of TRegionDefinition (detailed information can be found in the TRegionDefinition paragraph).
- **StylerName:** Holds the name of the styler.
- **Tag:** Is an integer property that has no predefined meaning. It can be used for storing an additional integer value, or it can be typecast to any value such as a component reference or a pointer.
- **Version:** Holds the component version number.

TElementStyles settings

When double clicking on the property, a string list editor is loaded, where TElementStyles can be added or removed:



Different TElementStyles can be added to the collection. In the example of the TTMSFMXMemoPascalStyler, following TElementStyles are needed:
0 – Pascal Standard Default: The keywords property holds all Delphi keywords.
1 – Simple Quote: Defines the simple quote character.
2 – Double Quote: Defines the double quote character.
3 – Symbols Delimiters: Defines the different symbols delimiters.

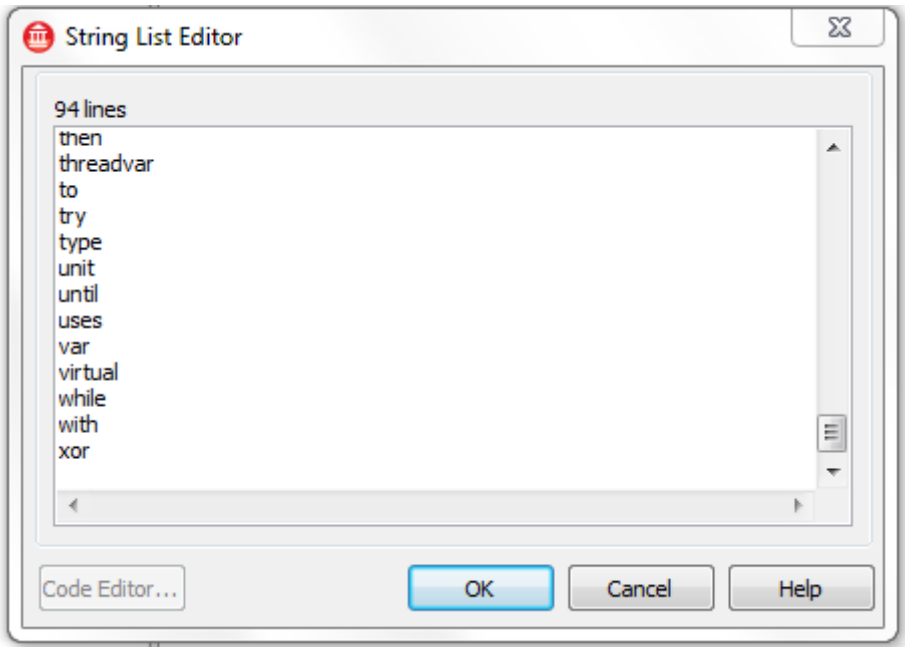
TElementStyles Properties

- **BracketEnd:** Holds the bracket end character, used when StyleType is stBracket. Note that when BracketEnd = #0, the end of the bracket style is considered when a word delimiter is found, i.e. a space or symbol.

- **BracketStart:** Holds the bracket start character, used when StyleType is stBracket.

~~TElementStyles.Properties: Pascal Standard Default:~~

- **BGColor:** Sets the background color used for the syntax element.
- **CommentLeft:** identifier string for start of a multiline comment
- **CommentRight**: identifier string for end of a multiline comment
- **Font:** Sets the font used for the syntax element.
- **Info:** Holds the name of the TElementStyle.
- **KeyWords:** Is a TStringlist holding all keywords. When double clicking on the property, a string list editor is loaded where keywords can be added or removed. This is only used when the the StyleType of the TElementStyle is stKeyword.



- **Symbols:** Holds the symbols when the StyleType is set to stSymbol.
- **StyleType:** Sets the StyleType to one of these values:
  - **stKeyWord:** keywords can be entered.
  - **stBracket:** bracket descriptions can be entered.
  - **stSymbol:** symbols can be entered.
  - **stComment:** comment left / right can be entered

TTMSFMXMemoBasicStyler

The TTMSFMXMemoBasicStyler is the syntax styler for MS Visual Basic files.
The properties are similar to the TTMSFMXMemoPascalStyler.

TTMSFMXMemoCSharpStyler

The TTMSFMXMemoCSharpStyler is the syntax styler for C-Sharp files.
The properties are similar to the TTMSFMXMemoPascalStyler.

TTMSFMXMemoCSSStyler

The TTMSFMXMemoCSSStyler is the syntax styler for CSS files.
The properties are similar to the TTMSFMXMemoPascalStyler.

TTMSFMXMemoHTMLStyler

The TTMSFMXMemoHTMLStyler is the syntax MemoStyler for HTML files.
The properties are similar to the TTMSFMXMemoPascalStyler.

TTMSFMXMemoJavaScriptStyler

The TTMSFMXMemoJavaScriptStyler is the syntax styler for JavaScript files.
The properties are similar to the TTMSFMXMemoPascalStyler.

~~TAdvJSMemoStyler properties~~

TTMSFMXMemoSQLMemoStyler

The TTMSFMXMemoSQLStyler is the syntax styler for SQL files.
The properties are similar to the TTMSFMXMemoPascalStyler.

TTMSFMXMemoWebMemoStyler

The TTMSFMXMemoWebStyler is the syntax styler for Web files (mix of HTML, CSS, JS).
The properties are similar to the TTMSFMXMemoPascalStyler.

TTMSFMXMemoXMLMemoStyler

The TTMSFMXMemoXMLStyler is the syntax styler for XML files.
The properties are similar to the TTMSFMXMemoPascalStyler.