Class Documentation: `TControl`

The `TControl` class is a fundamental class in the FiveWin library, which is used for creating and managing GUI controls in Windows applications. It inherits from the `TWindow` class and provides a wide range of functionalities for handling user interactions, resizing, dragging, and other control-related operations.

This class is designed to be the base class for various GUI controls like buttons, checkboxes, textboxes, etc. It includes methods and data members that allow for customization, event handling, and control manipulation.

Key Features

- **Drag and Drop Support**: The class supports dragging controls within the window, with grid alignment and resizing capabilities.
- **Event Handling**: It provides methods to handle mouse events, keyboard events, and focus changes.
- **Customizable Properties**: The class allows for customization of control properties such as size, position, alignment, and appearance.
- **Dynamic Resizing**: Controls can be resized dynamically, with support for relative and absolute positioning.
- **Transparency and 3D Look**: The class supports transparent controls and a 3D look for controls.

Data Members

Control State and Behavior

- `bSetGet`: A codeblock used to get or set the value of the control.
- `bChange`: A codeblock executed when the control's value changes.
- `bDragBegin`: A codeblock executed when dragging begins.
- `lCaptured`: Logical flag indicating whether the control has captured the mouse.
- `lDrag`: Logical flag indicating whether the control is in drag mode.
- `lMouseDown`: Logical flag indicating whether the mouse button is pressed on the control.
- `lUpdate`: Logical flag indicating whether the control should be updated.

- `l3DLook`: Logical flag indicating whether the control has a 3D look.
- `nLastRow`, `nLastCol`: Numeric values storing the last mouse position.
- `nMResize`: Numeric value indicating the type of mouse resizing (e.g., RES_NW, RES_N, etc.).
- `nDlgCode`: Numeric value returned by `GetDlgCode()` for default behavior.
- `oJump`: Reference to the next control to focus after validation.
- `l97Look`: Logical flag indicating whether the control has a Windows 97 look.
- `bSysKeyDown`: A codeblock executed when a system key is pressed.
- `bPreDelControl`, `bPostDelControl`: Codeblocks executed before and after deleting the control.
- `bDrag`, `bPostDrag`, `bPreDrag`: Codeblocks executed during and after dragging.
- `nClientBevel`: Numeric value indicating the bevel size of the client area.
- `lTransparent`: Logical flag indicating whether the control is transparent.
- `aSizeRect`: Array storing the size and position of the control.
- `ResizeType`: String indicating the type of resizing (e.g., "NW", "NE", etc.).
- `LastValidValue`: The last valid value of the control (read-only).
- `uOriginalValue`: The original value of the control (read-only).

Class Variables

- `nInitID`: Static numeric value used to generate unique control IDs.
- `nPoint`: Static numeric value used for mouse position tracking.
- `aDots`: Static array storing the dots used for resizing controls in design mode.
- `aProperties`: Static array listing the properties of the control (e.g., `cTitle`, `cVarName`, etc.).
- `bDelete`: Static codeblock executed when deleting a control.

Methods

Control Positioning and Sizing

- `nTop(nNewTop)`: Gets or sets the top position of the control.
- `nLeft(nNewLeft)`: Gets or sets the left position of the control.
- `AdjBottom()`: Adjusts the bottom position of the control.
- `AdjClient()`: Adjusts the client area of the control.
- `AdjLeft()`: Adjusts the left position of the control.
- `AdjRight()`: Adjusts the right position of the control.
- `AdjTop()`: Adjusts the top position of the control.

- `Move(nTop, nLeft, nWidth, nHeight, lRepaint)`: Moves the control to a new position and size.
- `CalcSize(nTop, nLeft, nWidth, nHeight, lRelative, oWnd, nBottom, nRight)`: Calculates the size and position of the control relative to its parent window.

Event Handling

- `Change()`: Virtual method called when the control's value changes.
- `ChangeFocus()`: Changes the focus to the next control.
- `Click()`: Handles the click event of the control.
- `LDblClick(nRow, nCol, nFlags)`: Handles the double-click event of the control.
- `LButtonDown(nRow, nCol, nKeyFlags, lTouch)`: Handles the left mouse button down event.
- `LButtonUp(nRow, nCol, nKeyFlags)`: Handles the left mouse button up event.
- `MouseMove(nRow, nCol, nKeyFlags)`: Handles the mouse move event.
- `KeyChar(nKey, nFlags)`: Handles the key press event.
- `KeyDown(nKey, nFlags)`: Handles the key down event.
- `KeyUp(nKey, nFlags)`: Handles the key up event.
- `SysChar(nKey, nFlags)`: Handles the system character event.
- `SysKeyDown(nKey, nFlags)`: Handles the system key down event.

Focus and Validation

- `GotFocus(hCtlLost)`: Handles the focus gain event.
- `LostFocus(hWndGetFocus)`: Handles the focus loss event.
- `lValid()`: Validates the control's value.
- `ForWhen()`: Evaluates the `WHEN` clause of the control.

Drag and Drop

- `DragBegin(nRow, nCol, nKeyFlags)`: Begins the drag operation.
- `MResize(nType, nRow, nCol, oDot)`: Resizes the control using the mouse.
- `ShowDots()`: Shows the resize dots around the control.
- `HideDots()`: Hides the resize dots.

Rendering and Appearance

- `Paint()`: Virtual method for painting the control.
- `PaintBack(hDC)`: Paints the background of the control.
- `EraseBkGnd(hDC)`: Erases the background of the control.

- `Colors(hDC)`: Sets the text and background colors of the control.
- `Set3DLook(lOnOff)`: Enables or disables the 3D look of the control.

Utility Methods

- `cToChar(cCtrlClass)`: Converts the control to a character representation.
- `GenLocals()`: Generates local variables for the control.
- `GetCtrlIndex()`: Returns the index of the control in its parent window.
- `Html()`: Returns an empty string (placeholder for HTML representation).
- `Init()`: Virtual method for initializing the control.
- `Initiate(hDlg)`: Initializes the control within a dialog.
- `Default()`: Sets default values for the control.
- `GetDlgCode(nLastKey)`: Returns the dialog code for the control.
- `GetNewId()`: Generates a new unique ID for the control.
- `Inspect(cDataName)`: Inspects the control's properties.
- `VarPut(uVal)`: Sets the value of the control.
- `VarGet()`: Gets the value of the control.

System Commands

- `SysCommand(nType, nLoWord, nHiWord)`: Handles system commands like Alt+key combinations.
- `HandleEvent(nMsg, nWParam, nLParam)`: Handles various Windows messages.

Additional Notes

> The class supports **grid-based resizing** and **dragging** of controls, which is useful in design mode.
>
> It provides **transparency** and **3D look** options for controls.
>
> The class is designed to work seamlessly with **FiveWin's event-driven architecture**, making it easy to handle user interactions.

Example Usage

```
// Create a new TControl instance
oControl := TControl():New()

// Set the control's position and size
oControl:Move(100, 100, 200, 50)

// Enable dragging
oControl:lDrag := .T.

// Handle the click event
oControl:bLClicked := { || MsgInfo("Control Clicked!") }

// Add the control to a window
oWnd:AddControl(oControl)
```

This class is a cornerstone of the FiveWin library, providing the foundation for creating interactive and customizable GUI controls in Windows applications.