

TableView
-log : ILog = LogManager.GetLogger(typeof(TableView))
-selectedCells : EventedList<TableViewCell>
-tableViewValidator : TableViewValidator
-isPasting : bool
-isSelectionChanging : bool
-updatingSelection : bool
-refreshRequired : bool
-refreshTimer : Timer
+TableViewContextMenu : ContextMenuStrip
-components : IContainer = null
-repositoryItemTimeEdit1 : RepositoryItemTimeEdit
#gridControl : GridControl
+gridView : GridView
<<Event>> +BeforeShowContextMenu : EventHandler
<<Event>> +FocusedRowChanged : EventHandler
<<Event>> +SelectionChangeStart : EventHandler
<<Event>> +SelectionChangeEnd : EventHandler
<<Event>> +UnboundColumnData : ColumnChangedDataHandler
<<Event>> +CellChanged : EventHandler<EventArgs<TableViewCell>>
<<Event>> +ColumnFilterChanged : EventHandler<ColumnChangedEventArgs>
<<Event>> +ColumnChanged : EventHandler<ColumnChangedEventArgs>
<<Event>> +ColumnHeaderClick : EventHandler<ColumnHeaderClickEventArgs>
<<Event>> +SelectedRowChanged : EventHandler<SelectedRowChangedEventArgs>
<<Event>> +SelectionChanged : EventHandler<TableSelectionChangedEventArgs>
<<Property>> +BindingContext : BindingContext
<<Property>> +ContextMenuStrip : ContextMenuStrip
<<Property>> +AllowDeleteRow : bool
<<Property>> +AllowColumnMoving : bool
<<Property>> +AllowAddNewRow : bool
<<Property>> +MultiSelect : bool
<<Property>> +IsEditing : bool
<<Property>> +RowSelect : bool
<<Property>> +IsSorted : bool
<<Property>> +ShowRowNumbers : bool
<<Property>> +ColumnAutoWidth : bool
<<Property>> +UseCenteredHeaderText : bool
<<Property>> +HideReadOnlyButtons : bool
<<Property>> +ReadOnly : bool
<<Property>> +MultipleCellEdit : bool
<<Property>> +AutoGenerateColumns : bool
<<Property>> +ColumnCount : int
<<Property>> +HeaderHeigh : int
<<Property>> +RowCount : int
<<Property>> +BestFitMaxRowCount : int
<<Property>> +SelectedDisplayRowIndex : int
<<Property>> +FocusedRowIndex : int
<<Property>> +SelectedRowIndex : int
<<Property>> +GetSelectedRows : int[]
<<Property>> +FilterTextExpression : string
<<Property>> +FilterTextSubstitute : string
<<Property>> +ExceptionMessageBoxAppendedText : string
<<Property>> +DataMember : string
<<Property>> +ReadOnlyCellForeColor : Color
<<Property>> +ReadOnlyCellBackColor : Color
<<Property>> +InvalidCellBackgroundColor : Color
<<Property>> +CurrentRowObject : object
<<Property>> +CurrentFocusedRowObject : object
<<Property>> +Data : object
<<Property>> +FocusedCell : TableViewCell
<<Property>> +Image : Image
<<Property>> +PasteBehaviour : TableViewPasteBehaviourOptions
<<Property>> +PasteController : ITableViewPasteController
<<Property>> +SelectedCells : IEventedList<TableViewCell>
<<Property>> +Columns : IList<ITableViewColumn>
<<Property>> +ColumnMenuItems : IList<TableViewColumnMenuItem>
<<Property>> +InvalidCellFilter : Func<TableViewCell, Boolean>
<<Property>> +ReadOnlyCellFilter : Func<TableViewCell, Boolean>
<<Property>> +DisplayCellFilter : Func<TableViewCellStyle, Boolean>
<<Property>> +RowValidator : Func<Int32, Object, RowValidationResult>[]
<<Property>> +InputValidator : Func<TableViewCell, Object, Tuple<String, Boolean>>
+TableView() -ConfigureContextMenu(): void -CreateRefreshTimer(): void -SubscribeEvents(): void +ResetBindings(): void +ClearColumns(): void +AddNewRowToDataSource(): void +EnsureVisible(item : object): void +OnNewRow(): bool +ClearSelection(): void +BestFitColumns(): void +GetRowIndexByDataSourceIndex(dataRowIndex : int): int +SetColumnEditor(tableColumnEditor : RepositoryItem, columnIndex : int): void +SetColumnFilter(columnName : string, filter : string): void +GetColumnFilter(columnName : string): string +BeginInit(): void +EndInit(): void +GetColumnByName(columnName : string): ITableViewColumn +AllowColumnSorting(allowed : bool): void +AllowColumnFiltering(allowed : bool): void +SelectRow(index : int): void +SelectRow(indices : int []): void +GetDataSourceIndexByRowIndex(rowIndex : int): int +RefreshData(): void +ScheduleRefresh(): void +CellsReadOnly(rowHandle : int, displayIndex : int): bool +SetCellValue(rowIndex : int, columnIndex : int, value : string): bool +SetRowCellValues(rowIndex : int, columnIndexDisplayStartIndex : int, cellValues : string []): bool +GetCellValue(rowIndex : int, columnIndex : int): object +SelectCells(top : int, left : int, bottom : int, right : int): void +PasteClipboardContents(): void +PasteLines(lines : string []): void +CopySelectionToClipboard(): void +SetFocus(rowIndex : int, columnIndex : int): void +DeleteCurrentSelection(): void +AddColumn(dataSourcePropertyName : string, columnCaption : string): void +AddColumn(dataSourcePropertyName : string, columnCaption : string, readOnly : bool, width : int): ITableViewColumn +GetCellDisplayText(rowIndex : int, columnIndex : int): string +AddUnboundColumn(columnName : string, t : Type, index : int, repositoryItem : RepositoryItem): int -GridViewDoubleClick(sender : object, e : EventArgs): void -GridViewCustomDrawCell(sender : object, e : RowCellCustomDrawEventArgs): void -GridViewColumnFilterChanged(sender : object, e : EventArgs): void -GridViewKeyUp(sender : object, e : KeyEventArgs): void -GridViewMouseUp(sender : object, e : MouseEventArgs): void -GridViewCellValueChanging(sender : object, e : CellValueChangedEventArgs): void -GridViewCustomDrawRowIndicator(sender : object, e : RowIndicatorCustomDrawEventArgs): void -GridViewShowGridMenu(sender : object, e : GridMenuEventArgs): void -GridView2RowCellStyle(sender : object, e : RowCellStyleEventArgs): void -GridViewShowingEditor(sender : object, e : CancelEventArgs): void -GridControlProcessGridKey(sender : object, e : KeyEventArgs): void -GridViewCustomFilterDisplayText(sender : object, e : ConvertEditValueEventArgs): void -GridViewFocusedColumnChanged(sender : object, e : FocusedColumnChangedEventArgs): void -GridViewClick(sender : object, e : EventArgs): void -GridViewCellValueChanged(sender : object, e : CellValueChangedEventArgs): void -GridViewSelectionChanged(sender : object, e : SelectionChangedEventArgs): void -GridViewFocusedRowChanged(sender : object, e : FocusedRowChangedEventArgs): void -GridViewCustomUnboundColumnData(sender : object, e : CustomColumnDataEventArgs): void -GridViewMouseEnter(sender : object, e : EventArgs): void -GridViewMouseDown(sender : object, e : MouseEventArgs): void -CopyPasteControllerPasteFailed(sender : object, e : EventArgs<string>): void -SelectedCellsCollectionChanged(sender : object, e : NotifyCollectionChangingEventArgs): void -DataSourceDataTableColumnsCollectionChanged(sender : object, e : CollectionChangeEventArgs): void -DataSourcePropertyCollectionChanged(sender : object, e : PropertyChangedEventArgs): void -DataSourceCollectionChanged(sender : object, e : NotifyCollectionChangingEventArgs): void -HandleHeaderPanelButton(sender : object, e : MouseEventArgs): void -GetItemByStringId(menu : DXPopupMenu, id : GridStringId): DXMenuItem -GetDataManager(): CurrencyManager -SubscribeToDataSource(): void -UnSubscribeFromDataSource(): void -RegionalSettingsManagerFormatChanged(): void -UpdateHeaderColumnSize(): void -GetSelectionsReadOnly(): bool -UpdateSelectionFromGridControl(): void -BuildColumnWrappers(): void -UpdateColumnsFormatting(): void -ShowEditorIfRowSelect(): void -GetColumn(displayIndex : int): GridColumn -UpdateColumnHeaderMenu(e : GridMenuEventArgs): void -IsNumberType(type : Type): bool -ValidateAndCommitRow(rowIndex : int): bool -SetCellValueInternal(rowIndex : int, columnIndex : int, value : string): bool -ConvertStringValueToDataValue(columnIndex : int, cellValue : string): object -ConvertToColumnValue(cellValue : string, columnType : Type): object #Dispose(disposing : bool): void -InitializeComponent(): void

XtraGridContextMenu
-log : ILog = LogManager.GetLogger(typeof (XtraGridContextMenu))
-components : IContainer = null
-btnUndo : ToolStripMenuItem
-ssdfToolStripMenuItem : ToolStripSeparator
-btnCopyAsString : ToolStripMenuItem
-btnCopyAsValue : ToolStripMenuItem
-btnPaste : ToolStripMenuItem
-btnDelete : ToolStripMenuItem
-sdfToolStripMenuItem : ToolStripSeparator
-btnFilter : ToolStripMenuItem
-btnAddRecord : ToolStripMenuItem
-btnDeleteRecord : ToolStripMenuItem
<<Property>> +SourceGrid : GridControl
<<Property>> -BindingSource : BindingSource
+XtraGridContextMenu() -XtraGridContextMenu_Closed(sender : object, e : ToolStripDropDownClosedEventArgs) : void -btnUndo_Click(sender : object, e : EventArgs) : void +CopySelectionAsValuesToClipboard(curGrid : GridView) : void +CopySelectionAsStringsToClipboard(curGrid : GridView) : void -btnCopyAsString_Click(sender : object, e : EventArgs) : void -btnCopyAsValue_Click(sender : object, e : EventArgs) : void -btnPaste_Click(sender : object, e : EventArgs) : void -btnDelete_Click(sender : object, e : EventArgs) : void -btnFilter_Click(sender : object, e : EventArgs) : void -btnAddRecord_Click(sender : object, e : EventArgs) : void -btnDeleteRecord_Click(sender : object, e : EventArgs) : void -XXXGridView_MouseDown(sender : object, e : MouseEventArgs) : void -XXXGridView_HiddenEditor(sender : object, e : EventArgs) : void -XXXGridView_ShownEditor(sender : object, e : EventArgs) : void -XXXGridView_CustomDrawCell(sender : object, e : RowCellCustomDrawEventArgs) : void -XXXGridView_GetFocus(sender : object, e : EventArgs) : void -XXXRepositoryItem_Spin(sender : object, e : SpinEventArgs) : void -XXXGridView_FocusedRowChanged(sender : object, e : FocusedRowChangedEventArgs) : void -XXXRepositoryItem_EditValueChanged(sender : object, e : EventArgs) : void -SetColumnValues(curGrid : ColumnView, dataRow : object, cols : string [], startCol : int) : void -GetColumnType(gridColumns : IList<GridColumn>, visibleIndex : int, propDesc : PropertyDescriptorCollection, columnHandle : int, bound : bool) : Type -ConvertToType(unboundType : UnboundColumnType) : Type -ConvertClipboardValueToDataValue(columnIndex : int, curGrid : ColumnView, cellValue : string, columnType : Type) : object +PasteClipboardContents() : void -PasteValues(curGrid : ColumnView, clipboardLines : string [], startRowIndex : int, startColIndex : int, endRowIndex : int, endColIndex : int) : int -SplitToCells(p : string) : string [] -GetClipboardLines() : string [] -GetStartRow(selection : GridCell []) : int -GetEndRow(selection : GridCell []) : int -GetStartColumn(selection : GridCell []) : int -GetEndColumn(selection : GridCell []) : int #Dispose(disposing : bool) : void -InitializeComponent() : void

TableViewColumnMenuItem
<<Event>> +Click : EventHandler <<Event>> +Showing : CancelEventHandler <<Property>> +InternalItem : DXMenuItem <<Property>> +Caption : string <<Property>> +Image : Image +TableViewColumnMenuItem(caption : string) -InternalItemClick(sender : object, e : EventArgs) : void +ShouldShow(column : ITableViewColumn) : bool

RectangleSelection
<<Property>> +Top : int <<Property>> +Left : int <<Property>> +Bottom : int <<Property>> +Right : int +RectangleSelection() +RectangleSelection(topLeft : TableViewCell, bottomRight : TableViewCell)



TableViewValidator
-tableView : TableView -originalRowValues : object[] -editingRow : bool -currentRowIndex : int <<Property>> +AutoValidation : bool +TableViewValidator(tableView : TableView) -BeginEditRow(rowIndex : int) : void -GetRowValues() : object [] -EndEditRow() : void -GetOriginalRowValues() : object [] +ValidateRow(view : GridView, rowIndex : int) : bool +ValidateCell(view : GridView, cell : TableViewCell, newValue : object, error : string) : bool +RefreshRowData() : void +RowLostFocus(sender : object, e : FocusedRowChangedEventArgs) : void +OnCellValueChanged(sender : object, e : CellValueChangedEventArgs) : void +OnValidateCell(sender : object, e : BaseContainerValidateEditorEventArgs) : void +OnValidateRow(sender : object, e : ValidateRowEventArgs) : void

TableViewContextMenu
-components : IContainer = null -btnPaste : ToolStripMenuItem -btnDelete : ToolStripMenuItem -btnCopy : ToolStripMenuItem <<Event>> +CopyClicked : EventHandler <<Event>> +PasteClicked : EventHandler <<Event>> +DeleteClicked : EventHandler <<Property>> +DeleteAvailable : bool <<Property>> +PasteAvailable : bool <<Property>> +DeleteEnabled : bool <<Property>> +PasteEnabled : bool <<Property>> +CopyEnabled : bool +TableViewContextMenu() -btnCopy_Click(sender : object, e : EventArgs) : void -btnPaste_Click(sender : object, e : EventArgs) : void -btnDelete_Click(sender : object, e : EventArgs) : void #Dispose(disposing : bool) : void -InitializeComponent() : void

RowValidationResult
<<Property>> +ColumnIndex : int <<Property>> +ErrorText : string <<Property>> +Valid : bool +RowValidationResult(errorText : string) +RowValidationResult(columnIndex : int, errorText : string)

TableViewPasteController
#log : ILog = LogManager.GetLogger(typeof(TableViewPasteController)) #tableView : ITableView <<Event>> +PasteFailed : EventHandler<EventArgs<String>> +TableViewPasteController(tableView : ITableView) +PasteLines(lines : string []) : void #GetPasteTargetSelection() : RectangleSelection #GetPasteTargetSelection(errorMessage : string) : RectangleSelection #GetRowSelectSelection() : RectangleSelection #GetRectangleSelection(cells : IList<TableViewCell>) : RectangleSelection +PasteClipboardContents() : void -OnPasteFailed(message : string) : void #CanPaste(clipboardLines : string [], errorMessage : string) : bool #PasteValues(clipboardLines : string [], targetSelection : RectangleSelection, addNewRows : bool) : RectangleSelection -PasteRowCellValuesAsARow(rowIndex : int, content : string [], pasteWidth : int, startColumnIndex : int) : void -PasteRowCellValuesOneByOne(j : int, content : string [], pasteWidth : int, startColumnIndex : int) : void #GetClipboardLines() : string [] #SplitToCells(p : string) : string []

TableViewArgumentBasedPasteController
<<Property>> +ArgumentColumns : IList<Int32> <<Property>> #NewRowsAddedToEnd : bool +TableViewArgumentBasedPasteController(tableView : ITableView, argumentColumns : IList<int>) #CanPaste(clipboardLines : string [], errorMessage : string) : bool -GetTargetColumns(clipboardLines : string [], targetSelection : RectangleSelection) : IEnumerable<int> -GetArgumentColumnNames() : string #PasteValues(clipboardLines : string [], targetSelection : RectangleSelection, addNewRows : bool) : RectangleSelection -PasteLine(line : string) : void -GetExistingRowForArguments(argumentValues : List<string>) : int

TableViewFunctionPasteController
+TableViewFunctionPasteController(tableView : ITableView, argumentColumns : IList<int>)

TableViewCellStyle
<<Property>> -AppearanceObject : AppearanceObject <<Property>> +ForeColor : Color <<Property>> +BackColor : Color <<Property>> +Selected : bool +TableViewCellStyle(appearanceObject : AppearanceObject, rowIndex : int, columnIndex : int, selected : bool)

TableViewExceptionHandler
-messageBoxAppendedText : string +TableViewExceptionHandler(messageBoxAppendedText : string) +GetLocalizedString(id : GridStringId) : string

TableViewColumn
-log : ILog = LogManager.GetLogger(typeof(TableViewColumn)) -dxGridView : GridView -tableView : ITableView -dxGridControl : GridControl -visibleIndex : int <<Property>> +DxColumn : GridColumn <<Property>> +Name : string <<Property>> +Caption : string <<Property>> +DisplayIndex : int <<Property>> +AbsoluteIndex : int <<Property>> +CustomFormatter : ICustomFormatter <<Property>> +DisplayFormat : string <<Property>> +ReadOnly : bool <<Property>> +Visible : bool <<Property>> +Editor : ITypeEditor <<Property>> +SortingAllowed : bool <<Property>> +FilteringAllowed : bool <<Property>> +IsFiltered : bool <<Property>> +FilterString : string <<Property>> +SortOrder : SortOrder <<Property>> +ColumnType : Type <<Property>> +ToolTip : string +TableViewColumn(view : GridView, control : GridControl, column : GridColumn, tableView : ITableView) +SetXtraGridDisplayFormat(value : string) : void -GetFormatType(displayFormat : string) : FormatType -RepositoryItemCloseUp(sender : object, e : CloseUpEventArgs) : void -RepositoryItemQueryCloseUp(sender : object, e : CancelEventArgs) : void -repositoryItem_Validating(sender : object, e : CancelEventArgs) : void -filePathTypeEditorRepositoryItem_ButtonClick(sender : object, e : ButtonPressedEventArgs) : void -buttonTypeEditorRepositoryItem_ButtonClick(sender : object, e : ButtonPressedEventArgs) : void -repositoryItem_QueryResultValue(sender : object, e : QueryResultValueEventArgs) : void -repositoryItem_QueryPopUp(sender : object, e : CancelEventArgs) : void -repositoryItem_Popup(sender : object, e : EventArgs) : void +Dispose() : void