

CELL ActiveX Help

Help Contents

1 ActiveX Interface

CELL Control is a Spreadsheet & Chart integrated ActiveX Control(Its previous name is OLE Control Component or OCX), the programming interface of ActiveX Control is made up of Property, Event and Method. Method is the essential for communicating with Control. Similar to Method , Property is a kind of Method displaying Control. Control uses Event interface to exchange information with users.

The programming interface of **CELL** Control is made up of:

Property

Method

Event

Because of the specialty of the **VBScript** data type (only variable data type), **CELL** Control especially provides some functions for implementing the Method and Event **in VBScript**. The Method used in **VBScript** is suffixed by “VB”, and for Event it is suffixed by “VBS”. It is convenient to use these interfaces in other developing kits.

If you want to know more information of interface’s name, please look up in the index.

2 How to use CELL Application

CELL Application is another independent program. It was supplied as CELL Control’s designer. CELL Application has same output file format with CELL Control, and CELL Control has open and save file interface, so you can design your form in CELL Application, save to a file, and open it in CELL Control. This can reduce code remarkably and make programming visual.

Property

<u>Cols</u>	the amount of the current sheet's rows
<u>Rows</u>	the amount of the current sheet's columns
<u>Border</u>	whether to display Control Component's border
<u>TopLabelVisible</u>	whether to display column label
<u>SideLabelVisible</u>	whether to display row label
<u>HScrollVisible</u>	whether to display to horizontal scroll bar's label
<u>VScrollVisible</u>	whether to display vertical roll bar's label
<u>PageLabelVisible</u>	whether to display page's label
<u>TopLabelHeight</u>	the height of column's label
<u>SideLabelWidth</u>	the width of row's label
<u>RowSelectedMode</u>	set the mode of selecting the whole row
<u>MultiSelectedMode</u>	set the mode of selecting the multiple rows
<u>DynamicEditArea</u>	whether allowing dynamically to expand edited area
<u>MultiColSizeMode</u>	whether allowing using mouse to alter multiple columns' width
<u>MultipleRowSizeMode</u>	whether allowing using mouse to alter multiple rows' width
<u>GridReadOnly</u>	set Control Component as read only mode
<u>EnablePopupMenu</u>	allow right key select menu
<u>CurrentTopLabelHint</u>	to highlight current CELL position in column label
<u>CurrentSideLabelHint</u>	to highlight current CELL position in row label
<u>ResizeWhenPasteNeed</u>	whether extend CELL when content pasted is beyond the boundary of cell
<u>AllowSizeCell</u>	whether allow drag CELL's corner using mouse
<u>EnableUndo</u>	whether allow UNDO
<u>AllowSizeColInGrid</u>	whether allow setting column's width in CELL
<u>AllowSizeRowInGrid</u>	whether allow setting row's height in CELL

Total 24 Property interfaces

Method

CELL Control Module provides a large number of Method accessing their own data. According to their functions, they can be classified as:

Print and Preview Method

Undo and Redo

CELL Basic Property

Cell Method

Cell Size Method

Drawing Method

Sheet Method

Row and Column Property MethodJoin cell Method

Formula and Expression

Chart Method

Cell Selection

Clipboard Method

Grid line Method

Search Replace and Locate Method

File and Database Method

Cell Position Method

Others

1 Print and Preview Method

<u>DoPrint</u>	print
<u>DoPrintPreview</u>	Overview what is to be printed .
<u>DoPrintPageSetup</u>	Set up page to be printed.
<u>DoSetPrintHead</u>	Set header.
<u>DoSetPrintFoot</u>	Set footnotes.
<u>DoSetPrintLabel</u>	Set row and column label to be printed.
<u>DoSetPrintTitle</u>	Set print title
<u>DoSetPrintPara</u>	Set print parameters(page, zoom etc.)
<u>DoSetPrintPara2</u>	Set print parameters(zoom, grid line etc.)
<u>DoSetPrinter</u>	Set printer
<u>DoPrintPage</u>	print page
<u>DoPrintPreviewPage</u>	print preview page

2 Undo and Redo

<u>DoUndo</u>	UNDO
<u>DORedo</u>	REDO
<u>DoDiscardUndo</u>	Discard UNDO Information

<u>DoGetUndoState</u>	Get current Undo state(Whether allow undo or not).
<u>DoGetRedoState</u>	Get current Redo state(Whether allow redo or not).

3 CELL Basic Property

<u>DoSetMessageTitle</u>	Set Message Box' title
<u>DoSetBackGround</u>	Set Control Component background mode
<u>DoSetDefaultFont</u>	Set font default font
<u>DoSetCanDragDrop</u>	Whether allow Drag and Drop
<u>DoSetUnScrollRow</u>	Set row none-rolling type
<u>DoSetUnScrollCol</u>	Set column none-rolling type
<u>IsModified</u>	the modified state of the contentin control
<u>DoSetModifiedFlag</u>	set modified state

4 Cell's Method

<u>DoSetCellAlignment</u>	Set cell alignment style
<u>DoSetCellTextStyle</u>	Set dell's text style
<u>DoSetCell3DState</u>	Set cell 3D style
<u>DoSetCellData</u>	Set cell data type
<u>DoSetPageCellData</u>	Set specified page's cell data type
<u>DoSetCellNumberStyle</u>	Set cell numeric style
<u>DoSetButtonCell</u>	Set cell button type
<u>DoSetButtonCellEx</u>	Set cell button type
<u>DoSetRadioCell</u>	Set cell radio button type
<u>DoSetDroplistCell</u>	Set cell drop list type
<u>DoSetCheckboxCell</u>	Set cell check box type
<u>DoSetTextSpinCell</u>	Set cell spin button type
<u>DoSetValueSpinCell</u>	Set cell value spin type
<u>DoSetNormalCell</u>	Set cell normal type
<u>DoSetCellInputControlCase</u>	Set cell control input text capital mode
<u>DoSetCellInputControlMask</u>	Set input control mask string
<u>DoSetCellInputOnlyValue</u>	Set cell only accept numeric
<u>DoSetCellColor</u>	Set cell's color
<u>DoSetCellPicture</u>	Set cell picture type
<u>DoGetCellData</u>	Get cell's data
<u>DoGetPageCellData</u>	Get cell's data in specified page
<u>DoSetCellValue</u>	Set cell's value
<u>DoSetCellString</u>	Set cell's string
<u>DoSetCellReadOnly</u>	Set cell read only type
<u>DoSetDropGridCell</u>	Set cell drop down window
<u>DoSetCellFont</u>	Set cell's font
<u>DoSetCellPrintable</u>	Set cell printable or not
<u>DoSetHintButtonCell</u>	set hint style button cell

5 Cell Size Method

<u>DoInsertCol</u>	Insert column
<u>DoInsertRow</u>	Insert row
<u>DoAppendCol</u>	Append column
<u>DoAppendRow</u>	Append row
<u>DoDeleteCol</u>	Delete column
<u>DoDeleteRow</u>	Delete row

6 Drawing Method

<u>DoRedrawAll</u>	Redraw all(including row label and column label)
<u>DoRedrawCell</u>	Redraw cell
<u>DoRedrawRange</u>	Redraw range
<u>DoRedrawGrid</u>	Redraw grid
<u>DoRedrawTopLabel</u>	Redraw column label
<u>DoRedrawSideLabel</u>	Redraw row label
<u>DoRedrawHScroll</u>	Redraw horizontal scroll bar
<u>DoRedrawVScroll</u>	Redraw vertical scroll bar
<u>DoRedrawPageLabel</u>	Redraw page label

7 Sheet Method

<u>DoDeletePage</u>	Delete page
<u>DoSetPageLabel</u>	Set page label
<u>DoAppendPage</u>	Append pages
<u>DoSetCurrentPage</u>	Set current page
<u>DoGetCurrentPage</u>	Get current page
<u>DoSetTotalPages</u>	Set amount of pages
<u>DoGetTotalPages</u>	Get amount of pages
<u>DoInsertPage</u>	Insert page
<u>DoCopyPage</u>	copy page(dest, source should exist)
<u>DoAppendPageFromFile</u>	append page from other file

8 Row and Column PropertyMethod

<u>DoSetDefaultRowHeight</u>	Set row height default
<u>DoSetDefaultColWidth</u>	Set column width default
<u>DoSetRowHeight</u>	Set row height
<u>DoSetColWidth</u>	Set column width
<u>DoSetRowHeightEx</u>	Set row height
<u>DoSetColWidthEx</u>	Set column Width
<u>DoGetColWidth</u>	Get column width
<u>DoGetColBestWidth</u>	Get most fitful column width
<u>DoGetRowHeight</u>	Get row height

<u>DoGetRowBestHeight</u>	Get most fitful row height
<u>DoSetEqualRowHeight</u>	Set row equal height
<u>DoGetColWidthVB</u>	Get column width
<u>DoGetRowHeightVB</u>	Get row height

9 Join cell Method

<u>DoJoinCells</u>	Set combined cells
<u>DoUnJoinCells</u>	Separate combined cells
<u>DoGetJoinRange</u>	Get the range of combined cells
<u>DoGetJoinRangeVB</u>	Get the range of combined cells

10 Formula and Expression

<u>DoSetFormula</u>	Set formula
<u>DoDelFormula</u>	Delete formula
<u>DoCalculateAll</u>	Calculate all pages
<u>DoCalculatePage</u>	Calculate one page
<u>DoIsFormulaCell</u>	Judge whether it is a formula cell
<u>DoAddUserFunction</u>	Add user defined function
<u>DoAddUserFunctionEx</u>	Add user defined function(can set default parameters)
<u>DoDelUserFunction</u>	Delete user defined function
<u>DoGetFormula</u>	Get formula
<u>DoCalculateExpr</u>	Calculate expression
<u>DoInputFormula</u>	Input formula
<u>DoFetchFuncParameter</u>	Get parameter value
<u>DoAddUserFunctionVB</u>	Add user defined function
<u>DoCalculateExprVB</u>	Calculate expression

11 Chart Method

<u>DoChartGuide</u>	Chart guide
<u>DoSetRefChart</u>	Set referenced chart
<u>DoSetChart</u>	Set chart
<u>DoSetChartRefData</u>	Set the Referenced data of chart
<u>DoSetChartData</u>	Set chart data
<u>DoDelChart</u>	Delete chart
<u>DoIsChartCell</u>	Judge whether it is chart cell
<u>DoSetChartValueData</u>	Set chart data
<u>DoSetChartStringData</u>	Set chart string
<u>DoSetChartGeneralData</u>	Set chart general data
<u>DoRefreshChart</u>	Refresh chart

12 Cell Selection

<u>DoSelectCell</u>	Select cell
---------------------	-------------

<u>DoSelectRange</u>	Select range
<u>DoGetSelectRange</u>	Get current selected range
<u>DoClearSelection</u>	Clear selection
<u>DoIsSelectedCell</u>	Judge whether cell is selected
<u>DoGetFirstSelectedCell</u>	Get the first selected cell
<u>DoGetNextSelectedCell</u>	Get the next selected cell
<u>DoShowCurrentCell</u>	Whether to highlight current cell
<u>DoGetFirstSelectedCellVB</u>	Get the first selected cell
<u>DoGetNextSelectedCellVB</u>	Get the next selected cell

13 Clipboard Method

<u>DoCopyArea</u>	Copy area
<u>DoCopySelected</u>	Copy selected area
<u>DoCutSelected</u>	Cut selected Cells
<u>DoCutArea</u>	Cut area
<u>DoPaste</u>	Paste

14 Grid Lines Method

<u>DoDrawHLine</u>	Draw horizontal line
<u>DoDrawVLine</u>	Draw vertical line
<u>DoDrawXLine</u>	Draw slash line
<u>DoDelXLine</u>	Delete cell line
<u>DoDelHLine</u>	Delete horizontal line
<u>DoDelVLine</u>	Delete vertical line
<u>DoDrawLineDlg</u>	Display DrawLine Dialog Box
<u>DoDrawLine</u>	Draw line
<u>DoClearLine</u>	Clear cell line

15 Search Replace and Locate Method

<u>DoFind</u>	Search
<u>DoReplace</u>	Replace
<u>DoLocate</u>	Locate
<u>DoShowFindDialog</u>	Show Search Dialog Box
<u>DoCloseFindDialog</u>	Close Find Dialog Box

16 File and Database Method

<u>DoOpenFileDbase</u>	Open file type database
<u>DoOpenFileDbaseVfp</u>	Open file type database
<u>DoOpenODBCDbase</u>	Open ODBC database
<u>DoOpenODBCDbaseVfp</u>	Open ODBC database
<u>DoDumpDbaseData</u>	Dump from database
<u>DoDumpDbaseDataVfp</u>	Dump from database

<u>DoDumpDbaseCell</u>	Dump from database
<u>DoCloseDbase</u>	Close Database
<u>DoSaveFile</u>	Save file
<u>DoOpenFile</u>	Open file
<u>DoRemoveODBCDSN</u>	Delete ODBC data source
<u>DoCreateODBCDSN</u>	Create ODBC data source
<u>DoSaveTextFile</u>	Output text file
<u>DoSaveHtmlFile</u>	Output HTML file
<u>DoReadTextFile</u>	Read text file

17 Cell Position Method

<u>DoGetLeftCol</u>	Get left column number
<u>DoGetTopRow</u>	Get top row number
<u>DoGetCurrentCol</u>	Get current cell's column
<u>DoGetCurrentRow</u>	Get current cell's row
<u>DoSetLeftCol</u>	Set left column
<u>DoSetTopRow</u>	Set top row
<u>DoMoveToCell</u>	Move current cell

18 Others

<u>DoSortByCol</u>	Sort by column
<u>DoSetSortCol</u>	Set columns to be sorted
<u>DoLogin</u>	Login
<u>GetUserDouble</u>	Get user numeric variable
<u>GetUserString</u>	Get user string variable
<u>SetUserDouble</u>	Set user numeric variable
<u>SetUserString</u>	Set user string variable
<u>AboutBox</u>	Version information

There are total 173 Method interfaces

Event

1 Allowed Operation

<u>OnAllowMove</u>	Allow to move current cell
<u>OnAllowMoveVBS</u>	Allow to move current cell
<u>OnAllowMoveToPage</u>	Change current page
<u>OnAllowSizeRow</u>	Set row's height
<u>OnAllowSizeRowVBS</u>	Set row's height
<u>OnAllowSizeCol</u>	Set column's width
<u>OnAllowSizeColVBS</u>	Set column's width
<u>OnAllowSizeTopLabel</u>	Set the height of column label
<u>OnAllowSizeTopLabelVBS</u>	Set the height of column label
<u>OnAllowSizeSideLabel</u>	Set the width of row label
<u>OnAllowSizeSideLabelVBS</u>	Set the width of row label
<u>OnAllowEditCell</u>	Edit cell
<u>OnAllowEditCellVBS</u>	Edit cell
<u>OnAllowInputFormula</u>	Input formula
<u>OnAllowInputFormulaVBS</u>	Input formula
<u>OnAllowChartGuide</u>	Chart guide
<u>OnAllowChartGuideVBS</u>	Chart guide
<u>OnAllowMoveToPage</u>	Change current page
<u>OnAllowDeleteCell</u>	Change current page
<u>OnAllowSizeRowInGrid</u>	In CELL (None row or column label grid) set row height
<u>OnAllowSizeColInGrid</u>	In CELL (None row or column label grid) set column width

2 Mouse and Keyboard Event

<u>OnLClickGrid</u>	Mouse left key clicks
<u>OnRClickGrid</u>	Mouse right key clicks
<u>OnDClickGrid</u>	Mouse left key double clicks
<u>OnLClickTopLabel</u>	Mouse left key clicks column label
<u>OnRClickTopLabel</u>	Mouse right key click column label
<u>OnDClickTopLabel</u>	Mouse left key double clicks column label
<u>OnLClickSideLabel</u>	Mouse left key clicks row label
<u>OnRClickSideLabel</u>	Mouse right key clicks row label
<u>OnDClickSideLabel</u>	Mouse left key double click row label
<u>OnLClickCorner</u>	Mouse left key clicks top left corner
<u>OnRClickCorner</u>	Mouse right key clicks top left corner
<u>OnDClickCorner</u>	Mouse left key double clicks top left corner
<u>OnKeyDown</u>	Press key
<u>OnKeyDownVBS</u>	Press key
<u>OnCharDown</u>	Input character
<u>OnCharDownVBS</u>	Input character

3Others

<u>OnNewPageSetup</u>	Set up new page
<u>OnColChange</u>	Change current column
<u>OnRowChange</u>	Change current row
<u>OnCellChange</u>	Change current cell
<u>OnLeftColChange</u>	Change leftmost column
<u>OnTopRowChange</u>	Change topmost row
<u>OnPageChange</u>	Change page
<u>OnCompareSortData</u>	Compare data when sorting
<u>OnCompareSortDataVBS</u>	Compare sorted data
<u>OnExecuteUserFunc</u>	Execute user defined function
<u>OnExecuteUserFuncVBS</u>	Execute user defined function
<u>OnUserFuncGuide</u>	Guide to user defined function
<u>OnGetCellData</u>	Get data of the cell
<u>OnGetCellDataVBS</u>	Get data of the cell
<u>OnEditError</u>	There is error when editing
<u>OnEditFinish</u>	Editing finish
<u>OnCellButtonClicked</u>	Occur when user click the button in a cell
<u>OnPrint</u>	Occur when user click the print button in print preview

There are total 55 event interfaces.

Cols

Set /Return the amount of current cell Control Component.

Syntax

object.Cols [= *number*]

Remarks

Cols Data member is a variable of long type, its range is 0-256.

Rows

Set /Return the property in current row numbers of cell **Control** Component.

Syntax

object.**Rows** [= *number*]

Remarks

Rows data value is a variable of **long** value, its range is 0-16384.

Border

Set /Return whether current cell in Control Component has boarder.

Syntax

object.**Border** [= *true/false*]

Remarks

Border data value is **BOOL** value.

TopLabelVisible

Set /Return whether the column label in current cell Control Component is visible.

Syntax

object.**TopLabelVisible** [= *true/false*]

Remarks

TopLabelVisible data value is **BOOL** value.

SideLabelVisible

Set /Return whether the column label in current cell Control Component is visible.

Syntax

object.**SideLabelVisible** [= *true/false*]

Remarks

SideLabelVisible data value is **BOOL** value.

HScrollVisible

Set /Return whether vertical scroll bar is visible when row number is beyond the displayed range in current cell Control Component

Syntax

object.HScrollVisible [= *true/false*]

Remarks

HScrollVisible data value is **BOOL** value.

VScrollVisible

Set /Return the value whether the horizontal scroll bar is visible when column number is beyond the displayed range in current cell Control Component

Syntax

object.VScrollVisible [= *true/false*]

Remarks

VScrollVisible data value is **BOOL** value.

PageLabelVisible

Set /Return the property of page label in current cell Control Component whether it is to be displayed.

Syntax

object.**PageLabelVisible** [= *true/false*]

Remarks

PageLabelVisible data value is **BOOL** value.

TopLabelHeight

Set /Return the height of column label in current cell Control Component.

Syntax

object.**TopLableHeight** [= *number*]

Remarks

TopLableHeight data value is **long** value.

SideLabelWidth

Set /Return the width of row label in current cell Control Component.

Syntax

object.SideLableHeight [= *number*]

Remarks

SideLableHeight data value is **long** value.

RowSelectedMode

Set /Return the value whether it is a selected row mode in current cell Control Component

Syntax

object.**RowSelectedMode** [= *true/false*]

Remarks

RowSlectedMode data value is **BOOL** value. If **RowSlectedMode** is *true*, click the grid, you will select the whole row of the grid it belongs to.

MultipleSelectedMode

Set /Return a value whether the grid is in a Multiple Selected mode in current cell Control Component.

Syntax

object.MutiSelectedMode [= *true/false*]

Remarks

MultipleSelectedMode data value is **BOOL** value. If **MultipleSelected Mode** is true, hold the key **Shift** or **Ctrl**, you can select more than one grid.

DynamicEditArea

Set /Return the value whether allow to dynamically extend the edited range while editing the contents of a grid in current cell Control Component.

Syntax

object.**DynamicEditArea** [= *true/false*]

Remarks

DynamicEditArea data value is **BOOL** value. If **DynamicEditArea** is *true*, edited range is beyond the width of the grid, it will automatically extend edited range. Otherwise, scroll edited contents automatically.

MultipleColSizeMode

Set /Return the property whether multiple columns' width is allowed to be changed together in current cell Control Component.

Syntax

object.MultipleColSizeMode [= *true/false*]

Remarks

MultipleColSizeMode data value is **BOOL** value.

MultipleRowSizeMode

Set /Return the property whether multiple rows' height is allowed to be changed together in current cell Control Component.

Syntax

object.**MultipleRowSizeMode** [= *true/false*]

Remarks

MultipleRowSizeMode data value is **BOOL** value.

EnablePopupMenu

Set /Return the property whether allow to popup right mouse key menu when click the right key of mouse in current cell Control Component.

Syntax

object.EnablePopupMenu [= *true/false*]

Remarks

EnablePopupMenu data value is **BOOL** value. Right key menu allow you to edit and print.

CurrentTopLabelHint

Set /Return whether it is allowed to display the column label the current grid belongs to in current cell Control Component.

Syntax

object. **CurrentTopLabelHint** [= *true/false*]

Remarks

CurrentTopLabelHint data value is **BOOL** value. If **CurrentTopLabelHint** is *true*, the column label of selected grid will change its state to selected.

CurrentSideLabelHint

Set /Return the property whether to display row number on row label in current cell Control Component.

Syntax

object. **CurrentSideLabelHint** [= *true/false*]

Remarks

CurrentSideLabelHint data value is **BOOL** value. If **CurrentSideLabelHint** *is true*, the row label of selected row will be concave.

ResizeWhenPasteNeed

Set /Return the property whether to pop up a query dialog when pasted contents is beyond the range of the cell in current cell Control Component.

Syntax

object. **ResizeWhenPasteNeed** [= *true/false*]

Remarks

ResizeWhenPasteNeed data value is **BOOL** value.

AllowSizeCell

Set /Return the property whether allow to resize grid using mouse dragging in current cell Control Component.

Syntax

object. **AllowSizeCell** [= *true/false*]

Remarks

AllowSizeCell data value is **BOOL** value.

GridReadOnly

Set /Return the property whether mark grid read only in current cell Control Component.

Syntax

object. **GridReadOnly** [= *true/false*]

Remarks

GridReadOnly data value is **BOOL** value. If **GridReadOnly** is *true*, you can not edit the contents of grid, you can not paste input formula and resize grid.

EnablePopupMenu

Set /Return the property whether right key menu is valid in current cell Control Component.

Syntax

object. **EnablePopupMenu** [= *true/false*]

Remarks

EnablePopupMenu data value is **BOOL** value.

CurrentTopLabelHint

Set /Return the property whether to show current cell position on concave column label in current cell Control Component.

Syntax

object. **CurrentTopLabelHint** [= *true/false*]

Remarks

CurrentTopLabelHint data value is **BOOL** value.

CurrentSideLabelHint

Set /Return the property whether to display current cell's position on concave row label in current cell Control Component.

Syntax

object. **CurrentSideLabelHint** [= *true/false*]

Remarks

CurrentSideLabelHint data value is **BOOL** value.

ResizeWhenPasteNeed

Set /Return the property whether to pop up query dialog box when paste is beyond the range in current cell Control Component.

Syntax

object. **ResizeWhenPasteNeed** [= *true/false*]

Remarks

ResizeWhenPasteNeed data value is **BOOL** value.

AllowSizeCell

Set /Return the property whether allow using mouse to drag the four corner of a cell in current cell
Control Component (Set combined cells)

Syntax

object. **AllowSizeCell** [= *true/false*]

Remarks

AllowSizeCell data value is **BOOL** value.

EnableUndo

Set /Return the property whether to allow UNDO while editing in current cell Control Component.

Syntax

object. **EnableUndo** [= *true/false*]

Remarks

EnableUndo data value is **BOOL** value.

AllowSizeColInGrid

Whether allow using mouse to set column width.

Syntax

object. **AllowSizeColInGrid** [= *true/false*]

Remarks

AllowSizeColInGrid data value is **BOOL** value. Default value is FALSE, that means that you are not allowed to set column width in a cell using mouse.

AllowSizeRowInGrid

Whether allow to set row height in a cell using mouse.

Syntax

object. **AllowSizeRowInGrid** [= *true/false*]

Remarks

AllowSizeRowInGrid data value is **BOOL** value. Default value is FALSE, that means you can not using mouse to set row height in a cell.

void DoPrint(BOOL showdlg)

Return Value:

None.

Parameters:

showdlg whether to display Print Set Dialog Box

Remarks:

Print cell.

void DoPrintPreview(BOOL allowPageset)

Return Value:

None.

Parameters:allowPageset

Whether allow to set Page Print Parameters.

Remarks:

Overview the contents to be printed.

void DoPrintPageSetup()

Return Value:

None.

Parameters:

None.

Remarks:

Pop up Page Setup Dialog Box.

**void DoSetPrintHead(LPCTSTR left, LPCTSTR mid, LPCTSTR right
)**

Return Value:

None.

Parameters:left left page header

mid middle page header

right right page header

Remarks:

Set page header

void DoSetPrintFoot(LPCTSTR left, LPCTSTR mid, LPCTSTR right)

Return Value:

None.

Parameters:left left footnotes

mid middle footnotes

right right footnotes

Remarks:

Set footnotes

void DoSetPrintLabel(BOOL collabel, BOOL rowlabel)

Return Value:

None.

Parameters:

collabel Print column label when it is true

rowlabel Print row label when it is true

Remarks:

Set row and column label to be printed

**void DoSetPrintTitle(long col1, long row1, long col2, long row2,
long rows, long cols)**

Return Value:

None.

Parameters:

col1, row1, col2, row2 range title use

rows row number row label belongs to

cols column number column label belongs to

Remarks:

Set the range of title, the row number when use row label, row number when use row label for printing(Mostly set the top of the title range to be title).

void DoSetPrintPara(float scale, short papersize, BOOL orientation);

Return Value:

None.

Parameters:

scale scaling, range between 0.1-4.0

papersize the size of paper, including:

- | | | |
|----|---------------------|-----------------------------|
| 1 | DMPAPER_LETTER | Letter 8 1/2 x 11 in |
| 2 | DMPAPER_LETTERSMALL | Letter Small 8 1/2 x 11 in |
| 3 | DMPAPER_TABLOID | Tabloid 11 x 17 in |
| 4 | DMPAPER_LEDGER | Ledger 17 x 11 in |
| 5 | DMPAPER_LEGAL | Legal 8 1/2 x 14 in |
| 6 | DMPAPER_STATEMENT | Statement 5 1/2 x 8 1/2 in |
| 7 | DMPAPER_EXECUTIVE | Executive 7 1/4 x 10 1/2 in |
| 8 | DMPAPER_A3 | A3 297 x 420 mm |
| 9 | DMPAPER_A4 | A4 210 x 297 mm |
| 10 | DMPAPER_A4SMALL | A4 Small 210 x 297 mm |
| 11 | DMPAPER_A5 | A5 148 x 210 mm |
| 12 | DMPAPER_B4 | B4 (JIS) 250 x 354 |
| 13 | DMPAPER_B5 | B5 (JIS) 182 x 257 mm |
| 14 | DMPAPER_FOLIO | Folio 8 1/2 x 13 in |
| 15 | DMPAPER_QUARTO | Quarto 215 x 275 mm |
| 16 | DMPAPER_10X14 | 10x14 in |
| 17 | DMPAPER_11X17 | 11x17 in |
| 18 | DMPAPER_NOTE | Note 8 1/2 x 11 in |
| 19 | DMPAPER_ENV_9 | Envelope #9 3 7/8 x 8 7/8 |
| 20 | DMPAPER_ENV_10 | Envelope #10 4 1/8 x 9 1/2 |
| 21 | DMPAPER_ENV_12 | Envelope #11 4 1/2 x 10 3/8 |
| 22 | DMPAPER_ENV_12 | Envelope #12 4 1/2 x 12 |
| 23 | DMPAPER_ENV_14 | Envelope #14 5 x 11 1/2 |
| 24 | DMPAPER_CSHEET | C size sheet |
| 25 | DMPAPER_DSHEET | D size sheet |
| 26 | DMPAPER_ESHEET | E size sheet |
| 27 | DMPAPER_ENV_DL | Envelope DL 110 x 220mm |
| 28 | DMPAPER_ENV_C5 | Envelope C5 162 x 229 mm |
| 29 | DMPAPER_ENV_C3 | Envelope C3 324 x 458 mm |
| 30 | DMPAPER_ENV_C4 | Envelope C4 229 x 324 mm |
| 31 | DMPAPER_ENV_C6 | Envelope C6 114 x 162 mm |
| 32 | DMPAPER_ENV_C65 | Envelope C65 114 x 229 mm |
| 33 | DMPAPER_ENV_B4 | Envelope B4 250 x 353 mm |
| 34 | DMPAPER_ENV_B5 | Envelope B5 176 x 250 mm |

35	DMPAPER_ENV_B6	Envelope B6 176 x 125 mm
36	DMPAPER_ENV_ITALY	Envelope 110 x 230 mm
37	DMPAPER_ENV_MONARCH	Envelope Monarch 3.875 x 7.5 in
38	DMPAPER_ENV_PERSONAL	6 3/4 Envelope 3 5/8 x 6 1/2 in
39	DMPAPER_FANFOLD_US	US Std Fanfold 14 7/8 x 11 in
40	DMPAPER_FANFOLD_STD_GERMAN	German Std Fanfold 8 1/2 x 12 in
41	DMPAPER_FANFOLD_LGL_GERMAN	German Legal Fanfold 8 1/2 x 13 in
42	DMPAPER_ISO_B4	B4 (ISO) 250 x 353 mm
43	DMPAPER_JAPANESE_POSTCARD	Japanese Postcard 100 x 148 mm
44	DMPAPER_9X14	9 x 11 in
45	DMPAPER_10X14	10 x 11 in
46	DMPAPER_15X14	15 x 11 in
47	DMPAPER_ENV_INVITE	Envelope Invite 220 x 220 mm
48	DMPAPER_RESERVED_48	RESERVED--DO NOT USE
49	DMPAPER_RESERVED_49	RESERVED--DO NOT USE
50	DMPAPER_LETTER_EXTRA	Letter Extra 9 1/2 x 12 in
51	DMPAPER_LEGAL_EXTRA	Legal Extra 9 1/2 x 15 in
52	DMPAPER_TABLOID_EXTRA	Tabloid Extra 11.69 x 18 in
53	DMPAPER_A4_EXTRA	A4 Extra 9.27 x 12.69 in
54	DMPAPER_LETTER_TRANSVERSE	Letter Transverse 8 1/2 x 11 in
55	DMPAPER_A4_TRANSVERSE	A4 Transverse 210 x 297 mm
56	DMPAPER_LETTER_EXTRA_TRANSVERSE	Letter Extra Transverse 9 1/2 x 12 in
57	DMPAPER_A_PLUS	SuperA/SuperA/A4 227 x 356 mm
58	DMPAPER_B_PLUS	SuperB/SuperB/A3 305 x 487 mm
59	DMPAPER_LETTER_PLUS	Letter Plus 8.5 x 12.69 in
60	DMPAPER_A4_PLUS	A4 Plus 210 x 330 mm
61	DMPAPER_A5_TRANSVERSE	A5 Transverse 148 x 210 mm
62	DMPAPER_B5_TRANSVERSE	B5 (JIS) Transverse 182 x 257 mm
63	DMPAPER_A3_EXTRA	A3 Extra 322 x 445 mm
64	DMPAPER_A5_EXTRA	A5 Extra 174 x 235 mm
65	DMPAPER_B5_EXTRA	B5 (ISO) Extra 201 x 276 mm
66	DMPAPER_A2	A2 420 x 594 mm
67	DMPAPER_A3_TRANSVERSE	A3 Transverse 297 x 420 mm
68	DMPAPER_A3_EXTRA_TRANSVERSE	A3 Extra Transverse 322 x 445 mm

orientation Orientation of the sheet, TRUE, portrait(PORTRAIT), FALSE, landscape(LANDSCAPE)

Remarks:

Set parameters of print. Note, if the pager size set is not supported by current printer, set it default, for example A4.

void DoSetPrintPara2(float scale, BOOL withgridline, BOOL allcell)

Return Value:

None.

Parameters:

scale scaling, range between 0.1-4.0

withgridline print gridline or not

allcell print all cell(some cells hide when print, but if you set allcell to TRUE, all cells will be printed).

Remarks:

This interface is suitable for special print request

void DoSetPrinter (LPCTSTR devicename)

Return Value:

None.

Parameters:

devicename printer name(note, case sensitive), less than 32 chars

Remarks:

The printer name won't save to file though you can set it's value.

void DoSetBackGround()

Return Value:

None.

Parameters:

style cell's boarder line style

Remarks:

Set the style of cell's boarder lines.

Style including:

- | | |
|---|----------------------|
| 0 | none boarder line |
| 1 | normal boarder line |
| 2 | convex boarder line |
| 3 | concave boarder line |

void DoSetCellAlignment(long col, long row, long align)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>align</i>	alignment mode

Remarks:

Set text alignment mode of cell.

If *col* or *row* is -1, the cell set is column label or row label.

Alignment mode is left;medium;right and the combination of top;medium;bottom.

The bits of *align* are as follows:

- bit0 = 1 left alignment
- bit1 = 1 right alignment
- bit2 = 1 vertical medium line alignment
- bit3 = 1 top alignment
- bit4 = 1 bottom alignment
- bit5 = 1 horizontal medium alignment

void DoSetCellTextStyle(long col, long row, long style)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>style</i>	style

Remarks:

Set text style of cell .

Style including:

0 normal display

Display text in single line. If the length of text is beyond the length of cell, only part of the text will be displayed.

1 multiple line display mode

Text is displayed in multiple lines.

2 abridgement display

When text is in single line mode, the part beyond cell's width is displayed by "...".

If *col* or *row* is -1, the cell set is column label or row label. If both are -1, the cell set is corner cell.

void DoSetCell3DState(long col, long row, long state)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>state</i>	3D state of cell

Remarks:

Set cell's display mode.

state value and cell's mode:

0	flat
1	convex
2	concave

If cell is in 1 or 2 state, the bottom boarder line of cell is invalid.

If *col* or *row* is -1, the cell set is column label or row label. If both are -1, the cell set is corner cell.

void DoSetCellData(long col, long row, const VARIANT FAR& data)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>data</i>	data to be set

Remarks:

Set data to cell.

If *col* or *row* is -1, the cell set is column label or row label. If both are -1, the cell set is corner cell.

void DoSetCellNumberStyle(long col, long row, BOOL commas, BOOL percent, short decimal, short currencychar, double unit, short minus, BOOL mathstyle)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>commas</i>	whether data is separated by comma from right to left in three digits step
<i>percent</i>	whether show per cent symbol
<i>decimal</i>	decimal digit number when display decimal
<i>currencychar</i>	currency symbol
<i>unit</i>	unit
<i>minus</i>	mode to display minus number
<i>mathstyle</i>	whether it is displayed in math mode

Remarks:

Set cell numeric display mode.

If *col* or *row* is -1, the cell set is column label or row label. If both are -1, the cell set is corner cell.

**void DoSetButtonCell(long col, long row, LPCTSTR text,
LPCTSTR mesg)**

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

text text on button

mesg message displayed when click button

Remarks:

Set current cell button cell.

void DoSetRadioCell(long col, long row, LPCTSTR items)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

items text of radio button

Remarks:

Set current cell radio button.

The texts of radio buttons are separated by “\n”.

**void DoSetDroplistCell(long col, long row, LPCTSTR items,
BOOL hidetag, BOOL readonly)**

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>items</i>	texts of droplist
<i>hidetag</i>	When selected cell is not current cell, whether hide droplist box
<i>readonly</i>	whether the item in drop list is editable

Remarks:

Set current cell drop list cell.

Items in drop list are separated by “\n”.

void DoSetCheckboxCell(long col, long row, LPCTSTR text)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>text</i>	text of check box

Remarks:

Set current cell check box cell.

void DoSetTextSpinCell(long col, long row, LPCTSTR items)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

items text of spin button

Remarks:

Set current cell spin button cell.

Items of spin buttons are separated by “\n”.

void DoSetValueSpinCell(long col, long row, double min, double max, double step)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>min</i>	minimum value of spin box
<i>max</i>	maximum value of spin box
<i>step</i>	increase step of data

Remarks:

Set current cell spin box cell.

void DoSetNormalCell(long col, long row)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

Remarks:

Set current cell normal cell.

void DoSetCellInputControlCase(long col, long row, short state)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>state</i>	letter in uppercase or lowercase

Remarks:

Set the letter mode in current cell.

state including:

- | | |
|---|--|
| 0 | capitalization is valid |
| 1 | convert letter from lowercase to uppercase |
| 2 | convert letter from uppercase to lowercase |

void DoSetCellInputControlMask(long col, long row, LPCTSTR mask)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

mask character mask, set the mode of inputted characters or digits of inputted numeric.

Remarks:

Set the maximum of total number of inputted characters or digits of inputted numeric of current cell.

Mask including:

0 -- numeric is needed

9 -- numeric and space

L -- English letters

l -- English letter and space

-- numeric and space , '+' , '-'

A -- letter and numeric

a -- letter numeric

& -- any printable character

To satisfied one's needs, one can combine *mask*.

void DoSetCellInputOnlyValue(long col, long row, BOOL maxflag, double max, BOOL minflag, double min)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>maxflag</i>	When inputted data is beyond maximum value, whether it is invalid and then indicates.
<i>max</i>	maximum value of inputted data
<i>minflag</i>	When inputted data is beyond minimum value, whether it is invalid and then indicates.
<i>min</i>	minimum value of inputted data

Remarks:

Set current cell to be data cell only accept numeric data.

void DoSetCellColor(long col, long row, long foreclr, long backclr)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

foreclr foreground color (**RGB**)

backclr background color (**RGB**)

Remarks:

Set the foreground color and background color of the cell .

If *col* or *row* is -1, the cell set is column label or row label. If both are -1, the cell set is corner cell. .

**void DoSetCellPicture(long col, long row, LPCTSTR picturefile,
long style)**

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>picturefile</i>	picture file
<i>style</i>	0 normal, 1 resize picture to fit cell

Remarks:

Insert picture into specified cell.

void DoInsertCol(long col, long count)

Return Value:

None.

Parameters:

col column number to insert columns

count total column number to be inserted

Remarks:

Insert a number of columns in front of specified column.

void DoInsertRow(long row, long count)

Return Value:

None.

Parameters:

row row number to insert row

count total number of rows to be inserted

Remarks:

Insert a number of rows in front of the specified row.

void DoAppendCol(long count)

Return Value:

None.

Parameters:

count total number of columns to be appended

Remarks:

Append a number of columns at the last column.

void DoAppendRow(long count)

Return Value:

None.

Parameters:

count total number of rows to be appended

Remarks:

Append a number of rows at the last row.

void DoRedrawAll()

Return Value:

None.

Parameters:

None.

Remarks:

Redraw cell **Control** Component.

void DoRedrawCell(long col, long row)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

Remarks:

Redraw current cell.

**void DoRedrawRange(long startcol, long startrow, long endcol,
long endrow)**

Return Value:

None.

Parameters:

<i>startcol</i>	column number of start column
<i>startrow</i>	row number of start row
<i>endcol</i>	column number of end column
<i>endrow</i>	row number of end row

Remarks:

Redraw all cells in selected range.

void DoRedrawGrid()

Return Value:

None.

Parameters:

None.

Remarks:

Redraw all cells.

void DoRedrawTopLabel()

Return Value:

None.

Parameters:

None.

Remarks:

Redraw all column label.

void DoRedrawSideLabel()

Return Value:

None.

Parameters:

None.

Remarks:

Redraw all row label.

void DoRedrawHScroll()

Return Value:

None.

Parameters:

None.

Remarks:

Redraw the horizontal scroll bar in cell Control Component.

void DoRedrawVScroll()

Return Value:

None.

Parameters:

None.

Remarks:

Redraw the vertical scroll bar in cell **Control** Component.

DoRedrawPageLabel()

Return Value:

None.

Parameters:

None.

Remarks:

Redraw page labels in cell **Control** Component.

void DoDeletePage(long index, long count)

Return Value:

None.

Parameters:

index page index of pages to be deleted

count amount of pages to be deleted

Remarks:

To delete a number of pages from specified page number.

void DoSetPageLabel(long index, LPCTSTR label)

Return Value:

None.

Parameters:

index page number index

label text of page label

Remarks:

Set the text of specified page label.

void DoDeleteCol(long pos, long count)

Return Value:

None.

Parameters:

pos the column to be deleted

count amount of columns to be deleted

Remarks:

Delete a number of columns from specified position.

DoDeleteRow(long pos, long count)

Return Value:

None.

Parameters:

pos position of the row to be deleted

count amount of rows to be deleted

Remarks:

Delete a number of rows from specified position.

void DoAppendPage(LPCTSTR label, long count)

Return Value:

None.

Parameters:

label page label of the pages appended

count amount of pages appended

Remarks:

Append a number of pages.

void DoSetCellFont(long col, long row, long size, long style, LPCTSTR name)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>size</i>	font size
<i>style</i>	font style
<i>name</i>	font name

Remarks:

Set font style of the text in specified cell.

Style values:

bit0 = 1 bold font

bit1 = 1 slash font

bit2 = 1 underline font

If *col* or *row* is -1, the cell set is column label or row label. If both are -1, the cell set is corner cell.

long DoGetLeftCol()

Return Value:

Column number of the first column in current visible range.

Parameters:

None.

Remarks:

Get the column number of the first column in current visible range.

long DoGetTopRow()

Return Value:

Row number of the first row in current visible range.

Parameters:

None.

Remarks:

Get the row number of the first row in current visible range.

long DoGetCurrentCol()

Return Value:

column number of the column where the cell located

Parameters:

None.

Remarks:

Get column number of currently selected cell.

long DoGetCurrentRow()

Return Value:

row number of the row where the cell located.

Parameters:

None.

Remarks:

Get row number of the currently selected cell.

void DoSetDefaultRowHeight(long height)

Return Value:

None.

Parameters:

height height of cell

Remarks:

Set the default height of cells in Control Component.

void DoSetDefaultColWidth(long width)

Return Value:

None.

Parameters:

width width of cell

Remarks:

Set the default width of cells in Control Component.

void DoSetRowHeight(long row, long height)

Return Value:

None.

Parameters:

row row number

height row height

Remarks:

Set the height of specified row in cell **Control** Component.

void DoSetColWidth(long col, long width)

Return Value:

None.

Parameters:

col column number

width column width

Remarks:

Set the width of specified column in cell **Control** Component.

void DoSetRowHeightEx(long row1, long row2, double height, short unit)

Return Value:

None.

Parameters:

row1, row2 range of rows

height row height

unit height unit, 0 – pixel 1 – centimeter

Remarks:

Set the height of specified row in cell **Control** Component.

void DoSetColWidthEx(long col1, long col2, double width, short unit)

Return Value:

None.

Parameters:

col1, col2 the range of columns

width column width

unit width unit, 0 – pixel 1 – centimeter 2 – number of characters

Remarks:

Set width in specified column in cell **Control** Component.

BOOL DoGetColWidth(long col, long FAR* width)

Return Value:

If there is column label, return **TRUE**, else return **FALSE**.

Parameters:

col column number

width width(in pixel)

Remarks:

Get the width of specified column.

long DoGetColBestWidth(long col)

Return Value:

If there is column label, return the most fitful width, else return **0**.

Parameters:

col column number

width width(in pixel)

Remarks:

Get the most fitful width in specified column.

BOOL DoGetRowHeight(long row, long FAR* height)

Return Value:

If there is row label, return **TRUE**, else return **FALSE**.

Parameters:

col column

height height(in pixel)

Remarks:

Get the height of specified row.

long DoGetRowBestHeight(long row)

Return Value:

If there is row label, return the most fitful height, else return **0**.

Parameters:

col row number

width height(in pixel)

Remarks:

Get the most fitful height of specified row.

void DoSortByCol(BOOL direction, long col)

Return Value:

None.

Parameters:

<i>direction</i>	the direction of sorting
<i>col</i>	column number

Remarks:

Set the direction of sorting in specified column.

If *direction* is 1, the mode is ascending. If it is 0, the mode is descending mode.

If the specified column can not be sorted, it is invalid.

Note the combined cell; formula and graph in the column will be deleted.

void DoSetSortCol(long col, BOOL sort)

Return Value:

None.

Parameters:

col column number

sort whether allow sorting. If it *is true*, sorting is allowed. Else, sorting is disabled.

Remarks:

Set whether allow specified column to be sorted.

void DoJoinCells(long startcol, long startrow, long endcol, long endrow)

Return Value:

None.

Parameters:

startcol start column

startrow start row

endcol end column

endrow end row

Remarks:

Combine the cells in specified range.

void DoUnJoinCells(long col, long row)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

Remarks:

Separate the cells in specified range.

This method deals only with combined cells, with no basic cell.

void DoSetLeftCol(long col)

Return Value:

None.

Parameters:

col column number

Remarks:

Put the specified column to the left of visible range.

void DoSetTopRow(long row)

Return Value:

None.

Parameters:

row row number

Remarks:

Put the specified row to the top of visible range.

void DoMoveToCell(long col, long row)

Return Value:

None.

Parameters:

col number of the column cell locates

row number of the row cell locates

Remarks:

Move the selected range to the specified cell.

**BOOL DoGetJoinRange(long col, long row, long FAR* startcol,
long FAR* startrow, long FAR* endcol, long FAR* endrow)**

Return Value:

If succeeds, return **TRUE**. Else, return **FALSE**.

Parameters:

<i>col</i>	number of column the cell locates
<i>row</i>	number of row the cell locates
<i>startcol</i>	pointer to the column where range begins
<i>startrow</i>	pointer to the row where range begins
<i>endcol</i>	pointer to the column where range ends
<i>endrow</i>	pointer to the row where range ends

Remarks:

Get the position information of specified combined cells.

void DoSetFormula(long col, long row, LPCTSTR text)

Return Value:

None.

Parameters:

<i>col</i>	column number
<i>row</i>	row number
<i>text</i>	formula expression string

Remarks:

Set specified cell property formula.

Formula can be not only the standard formula in cell **Control** Component, but also user defined formula.

void DoDelFormula(long col, long row)

Return Value:

None.

Parameters:

col column number

row row number

Remarks:

Delete the formula in specified cell.

void DoCalculateAll()

Return Value:

None.

Parameters:

None.

Remarks:

Calculate all formula cells value in the cell.

BOOL DoIsFormulaCell(long col, long row)

Return Value:

If current cell is formula cell, return **TRUE**. If it is other type, return **FALSE**.

Parameters:

col column number

row row number

Remarks:

Judge whether specified cell is formula cell or not.

void DoSetRefChart(long col, long row, long startcol, long startrow, long endcol, long endrow)

Return Value:

None.

Parameters:

<i>col</i>	column number of graphic cell
<i>row</i>	row number of graphic cell
<i>startcol</i>	start column in referenced range
<i>startrow</i>	start row in referenced range
<i>endcol</i>	end column in referenced range
<i>endrow</i>	end row in referenced range

Remarks:

Set the data source range of specified graphic cell.

void DoSetChart(long col, long row, long datacols, long datarows)

Return Value:

None.

Parameters:

<i>col</i>	column number of the cell
<i>row</i>	row number of the cell
<i>datacols</i>	amount of graphic data column
<i>datarows</i>	amount of graphic data row

Remarks:

Set specified cell a new specified graphic cell size.

**void DoSetChartRefData(long col, long row, long chartdatacol,
long chartdatarow, long refcol, long refrow)**

Return Value:

None.

Parameters:

<i>col</i>	column number of graphic cell
<i>row</i>	row number of graphic cell
<i>chartdatacol</i>	column number of data cell in chart
<i>chartdatarow</i>	row number of data cell in chart
<i>refcol</i>	referenced cell column number of data source in chart's data cell
<i>refrow</i>	referenced cell row number of data source in chart's data cell

Remarks:

Set the source data cell of the specified chart cell's data cell.

void DoSetChartData(long col, long row, long chartdatacol, long chartdatarow, const VARIANT FAR& data)

Return Value:

None.

Parameters:

<i>col</i>	column number of the chart where the cell located
<i>row</i>	row number of the chart where the cell located
<i>chartdatacol</i>	column number of data cell in chart cell
<i>chartdatarow</i>	row number of data cell in chart cell
<i>data</i>	data

Remarks:

Set data to the data cell in specified chart cell.

void DoDelChart(long col, long row)

Return Value:

None.

Parameters:

col column number of char cell

row row number of char cell

Remarks:

Delete specified chart cell.

BOOL DoIsChartCell(long col, long row)

Return Value:

If the cell is a chart cell, return **TRUE**, else return **FALSE**

Parameters:

<i>col</i>	column number of chart cell
<i>row</i>	row number of chart cell

Remarks:

Judge whether specified cell is a chart cell.

void DoSetDefaultFont(long size, long style, LPCTSTR name)

Return Value:

None.

Parameters:

size font size

style font style

name font name

Remarks:

Set the default font to cell.

Style value:

0 bold font

1 slash font

2 underline font

void DoSelectCell(long col, long row)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

Remarks:

Set specified cell selected state.

**void DoSelectRange(long startcol, long startrow, long endcol,
long endrow)**

Return Value:

None.

Parameters:

<i>startcol</i>	column number of start column in specified range
<i>startrow</i>	row number of start row in specified range
<i>endcol</i>	column number of end column in specified range
<i>endrow</i>	row number of end row in specified range

Remarks:

Set cells in specified range selected state.

BOOL DoGetSelectRange(VARIANT FAR* startcol, VARIANT FAR* startrow, VARIANT FAR* endcol, VARIANT FAR* endrow)

Get current selected range.

Return Value: True Currently there is selected range, False Currently there is no selected range.

Parameters:

<i>startcol</i>	column number of start column in specified range
<i>startrow</i>	row number of start row in specified range
<i>endcol</i>	column number of end column in specified range
<i>endrow</i>	row number of end row in specified range

Remarks:

If there are multiple selected ranges , it refers to the last one.

void DoClearSelection()

Return Value:

None.

Parameters:

None.

Remarks:

Clear all in all selected cell.

BOOL DoIsSelectedCell(long col, long row)

Return Value:

If current cell is selected, return TRUE, else return FALSE.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located

Remarks:

Judge whether specified cell is selected.

void DoCopyArea(long startcol, long startrow, long endcol, long endrow)

Return Value:

None.

Parameters:

<i>startcol</i>	column number of start column in specified range
<i>startrow</i>	row number of start row in specified range
<i>endcol</i>	column number of end column in specified range
<i>endrow</i>	row number of end row in specified range

Remarks:

Copy the data of specified range to paste board.

void DoCopySelected()

Return Value:

None.

Parameters:

None.

Remarks:

Copy the data of specified range to paste board.

void DoCutSelected()

Return Value:

None.

Parameters:

None.

Remarks:

Cut the data of specified range and then copy them to paste board.

void DoCutArea(long startcol, long startrow, long endcol, long endrow)

Return Value:

None.

Parameters:

<i>startcol</i>	column number of start column in specified range
<i>startrow</i>	row number of start row in specified range
<i>endcol</i>	column number of end column in specified range
<i>endrow</i>	row number of end row in specified range

Remarks:

Cut the data of specified range and then copy them to paste board.

DoPaste(long col, long row, BOOL samesize)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>samesize</i>	whether they are the same size

Remarks:

Paste data of paste board to specified cell.

If *samesize* is TRUE, paste data originally. Else, change pasted data to comply with the size of current cell.

BOOL DoGetFirstSelectedCell(long FAR* col, long FAR* row)

Return Value:

If succeed return TRUE, else return FALSE

Parameters:

<i>col</i>	pointer to column number of cell
<i>row</i>	pointer to row number of cell

Remarks:

Get the position of the first cell in selected cells.

DoGetNextSelectedCell(long FAR* col, long FAR* row)

Return Value:

If succeed, return TRUE, else return FALSE

Parameters:

col pointer to column number of cell

row pointer to row number of cell

Remarks:

Get the position of the next cell in selected cells.

BOOL DoGetCellData(long col, long row, VARIANT FAR* data)

Return Value:

If succeeds, return TRUE, else return FALSE

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>data</i>	returned data pointer

Remarks:

Get data of specified cell.

void DoDrawHLine(long col1, long col2, long row, BOOL top, long color, long size)

Return Value:

None.

Parameters:

<i>col1</i>	column number of start column
<i>col2</i>	column number of end column
<i>row</i>	row number of the cell
<i>top</i>	if TRUE , draw lines of top boarder£»if FALSE , draw lines of bottom boarder.
<i>color</i>	color of lines (RGB)
<i>size</i>	size of lines (0 – none 1 – slim line 2 – medium line 3 – thick line)

Remarks:

Draw top boarder or bottom boarder line of specified cells.

DoDrawVLine(long row1, long row2, long col, BOOL left, long color, long size)

Return Value:

None.

Parameters:

row1 row number of start row

row2 row number of end row

col column number of the cell

left If **TRUE**, draw lines of left boarder£»if **FALSE**, draw lines of right boarder.

color color of lines (**RGB**)

size size of lines (0 – none 1 – slim line 2 – medium line 3 – thick line)

Remarks:

Draw left boarder or right boarder line of cells in specified range.

**void DoDrawXLine(long col1, long row, long col2, long row2,
BOOL dir, long color, long size)**

Return Value:

None.

Parameters:

col1 column number of start column

row row number of start row

col2 column number of end column

row2 row number of end row

dir if TRUE, direction is from top left to bottom right£»if FALSE, direction is from
bottom left to top right

color color of lines (**RGB**)

size size of lines (0 – none 1 – slim line 2 – medium line 3 – thick line)

Remarks:

Draw diagonal line specified range.

col2 and *row2* must not be less than *col1* and *row1*, or it is invalid.

DoDelXLine(long col1, long row1, long col2, long row2)

Return Value:

None.

Parameters:

col1 column number of start column

row1 row number of start row

col2 column number of end column

row2 row number of end row

Remarks:

Delete the diagonal line in specified range.

void DoDelUserFunction(LPCTSTR name)

Return Value:

None.

Parameters:

name name of user defined function

Remarks:

Delete user defined function expression.

DoDelHLine(long col1, long col2, long row, BOOL top)

Return Value:

None.

Parameters:

col1 column number of start column

col2 column number of end column

row row number of the columns

top If it is **TRUE**, delete top boarder lines of cells£»else if it is **FALSE**, delete bottom

boarder lines of cells.

Remarks:

Delete top boarder lines or bottom boarder lines of specified cells.

void DoDelVLine(long row1, long row2, long col, BOOL left)

Return Value:

None.

Parameters:

row1 row number of start row

row2 row number of end row

row row number of columns

left If it is **TRUE**, delete left boarder lines of cellsE»else if it is **FALSE**, delete

right boarder lines.

Remarks:

Delete left boarder or right boarder lines of specified cells.

BOOL DoAddUserFunction(LPCTSTR functype, LPCTSTR name, short rtype, short par anum, short FAR* paratype, LPCTSTR funcelpstring)

Return Value:

If add successfully, return TRUE. else return FALSE

Parameters:

<i>functype</i>	function type
<i>name</i>	name of function
<i>rtype</i>	return value type of function, valid value: 0, result is numeric£»1, result is string£»4, result is string or numeric.
<i>par anum</i>	number of parameter
<i>paratype</i>	pointer to the type of parameter
<i>funcelpstring</i>	information about function

Remarks:

Add a user defined function.

Paratype values:

0	numeric(float type)
1	string
2	cell
3	range
4	numeric or string

e.g. a function with 3 parameters, they are numeric, string and cell, then paratype point to array: { 0, 1, 2 }.

BOOL DoAddUserFunctionEx(LPCTSTR functype, LPCTSTR name, short rtype, short parnum, short FAR* paratype, short FAR* defaultpara, LPCTSTR funchelpstring)

Return Value:

If add successfully, return TRUE. else return FALSE

Parameters:

<i>functype</i>	function type
<i>name</i>	name of function
<i>rtype</i>	return value type of function, valid value: 0, result is numeric£»1, result is string£»4, result is string or numeric.
<i>paranum</i>	number of parameter
<i>paratype</i>	pointer to the type of parameter
<i>defaultpara</i>	indicate parameters can be absent, short array
<i>funchelpstring</i>	information about function

Remarks:

Add a user defined function.

Paratype is an array, valid value of the item in array:

0	numeric(float type)
1	string
2	cell
3	range
4	numeric or string

e.g. a function with 3 parameters, they are numeric, string and cell, then paratype point to array: { 0, 1, 2 }.

defaultpara is an array, valid value of the item in array:

- 0, indicate this parameter hasn't default value, can't be absent
- 1, indicate this parameter has default value, can be absent

e.g. suppose the last two parameters can be absent, then defaultpara would point to array: { 0, 1, 1 }.

void DoSetCurrentPage(long pageno)

Return Value:

None.

Parameters:

pageno page number

Remarks:

Set specified page number page the current page.

long DoGetCurrentPage()

Return Value:

Current page number, start from 0.

Parameters:

None.

Remarks:

Get page number of current page.

void DoSetTotalPages(long pages)

Return Value:

None.

Parameters:

pages amount of pages

Remarks:

Set the total of pages in Control Component.

long DoGetTotalPages()

Return Value:

The total of pages in cell.

Parameters:

None.

Remarks:

Get the the total of pages in cell.

void DoSetCanDragDrop(BOOL drag)

Return Value:

None.

Parameters:

drag whether allow drag-drop

Remarks:

Set the property whether allow drag-drop in a cell.

void DoSetUnScrollRow(long row1, long row2)

Set unscrolling rows

Parameters:

row1, row2 row number of start row and end row with no scrolling. When they are -1, clear current non-scroll rows (set them to normal)

Remarks:

The total of non-scrolling rows is 10.

void DoSetUnScrollCol(long col1, long col2)

Set non-scrolling columns

Parameters:

col1, col2 column number of non-scrolling start column and end column. If they are both -1, clear non-scrolling column property. (Set them to normal)

Remarks:

The total of non-scrolling column is 10.

void DoInsertPage(LPCTSTR label, long location, long count)

Return Value:

None.

Parameters:

<i>label</i>	page label
<i>location</i>	inserting location
<i>count</i>	total of pages to be insert

Remarks:

Insert a number of pages in specified location.

void DoShowCurrentCell(BOOL focus, BOOL highlight)

Return Value:

None.

Parameters:

focus whether there is focus input box

highlight whether highlight what is displayed

Remarks:

Determine the display mode of current cell.

void DoSetEqualRowHeight(BOOL equal)

Return Value:

None.

Parameters:

equal whether all rows in the cell have the same height.

Remarks:

Set the height of all rows in the cell to be equal or not.

void DoChartGuide(long col, long row, long col2, long row2)

Return Value:

None.

Parameters:

col start column

row start row

col2 end column

row2 end row

Remarks:

Set specified range chart cell, and pop up chart guide dialog box.

void DoInputFormula(long col, long row)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

Remarks:

Set specified cell formula cell, and pop up input formula dialog box.

BOOL DoFetchFuncParameter(short no, VARIANT FAR* para)

Return Value:

If succeed, return TRUE, else return FALSE.

Parameters:

<i>no</i>	index of parameter value(note! Not parameter index)
<i>para</i>	pointer to parameter value

Remarks:

Get user defined function's parameters' value.

When an user defined function is evaluated, event OnExecuteUserFunc occurs, you can use this method to get user defined function's parameters' value in this event.

Note: there are numeric, string, cell, and range type parameters, when parameter is a cell, it will expand to 2 VARIANT variable, when parameter is a range, it will expand to 4 VARIANT variable, so index of parameter value is not user defined function's parameter index, you can see this in VCDEMO.

Note: If a parameter is default(absent), it's correspond value, the VARIANT variable's type will be Empty(0, you can get by VarType() function in VB).

void DoLogin(long id, LPCTSTR passwd)

Return Value:

None.

Parameters:

id login identification

passwd login password

Remarks:

User login.

Authorized cell Control Module users can get their user identification and the associated password. Please call this method after construct a cell object.

BOOL DoCalculateExpr(LPCTSTR expr, short FAR* rettype, VARIANT FAR* exprresult)

Return Value:

If succeed in getting the value, return **TRUE**, else return **FASLE**.

Parameters:

<i>expr</i>	expression string
<i>rettype</i>	pointer to the return expression type
<i>exprresult</i>	pointer to the return result

Remarks:

Calculate specified expression's value.

If return type is 1, return character string. If return type is 0, return value.

void DoSetCellValue(long col, long row, double value)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

value value set

Remarks:

Set specified cell double dimension float number.

If *col* or *row* is -1, the cell set is column label or row label. If they both are -1, the cell set is corner cell.

void DoSetCellString(long col, long row, LPCTSTR string)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

string character string set

Remarks:

Set character string to specified cell.

If *col* or *row* is -1, the cell set is column label or row label. If they both are -1, the cell set is corner cell.

void DoSetChartGeneralData(long col, long row, short charttype, BOOL xaxisAttr, LPCTSTR title, LPCTSTR foot, LPCTSTR yaxisTitle)

Return Value:

None.

Parameters:

<i>col</i>	column number of the chart where the cell located
<i>row</i>	row number of the chart where the cell located
<i>charttype</i>	chart type
<i>xaxisAttr</i>	attribute of X axis (reference chart guide)
<i>title</i>	chart title
<i>foot</i>	chart footnotes
<i>yaxisTitle</i>	title of Y axis

Remarks:

Set the chart of the chart cell.

void DoRefreshChart(long col, long row)

Return Value:

None.

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

Remarks:

Refresh the chart in specified chart cell.

BOOL DoGetFormula(long col, long row, VARIANT FAR* text)

Return Value:

If succeed, return TRUE, else return FALSE.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>text</i>	pointer to expression

Remarks:

Get the formula expression in formula cell.

void DoDrawLineDlg()

Return Value:

None.

Parameters:

None.

Remarks:

Pop up cell line setup dialog box.

void DoDrawLine(long startcol, long startrow, long endcol, long endrow, long type, long size, long color)

Return Value:

None.

Parameters:

<i>startcol</i>	start column
<i>startrow</i>	start row
<i>endcol</i>	end column
<i>endrow</i>	end row
<i>type</i>	type
<i>size</i>	size of the line
<i>color</i>	color of the line

Remarks:

Draw cell line in specified range.

Values of *type*:

0	all lines in selected range
1	boarder line of selected range
2	left boarder line of the cells in selected range
3	right boarder line of the cells in selected range
4	top boarder line of the cells in selected range
5	bottom boarder line of the cells in selected range
6	slash line of the cells in selected range
7	reverse slash line of the cells in selected range

**void DoSetChartData(long col, long row, long chartcol,
long chartrow, double value)**

Return Value:

None.

Parameters:

<i>col</i>	column number of the chart where the cell located
<i>row</i>	row number of the chart where the cell located
<i>chartcol</i>	column number of the data cell in chart where the cell located
<i>chartrow</i>	row number of the data cell in chart where the cell located
<i>value</i>	value set

Remarks:

Set double dimension value to the chart specified cell in chart cell.

**void DoSetChartStringData(long col, long row, long chartcol,
long chartrow, const VARIANT FAR& text)**

Return Value:

None.

Parameters:

<i>col</i>	column number of the chart where the cell located
<i>row</i>	row number of the chart where the cell located
<i>chartcol</i>	column number of the data cell in chart where the cell located
<i>chartrow</i>	row number of the data cell in chart where the cell located
<i>text</i>	character string set

Remarks:

Set character string to specified cell in chart cell.

void DoClearLine(long startcol, long startrow, long endcol, long endrow, long type)

Return Value:

None.

Parameters:

<i>startcol</i>	start column
<i>startrow</i>	start row
<i>endcol</i>	end column
<i>endrow</i>	end row
<i>type</i>	type

Remarks:

Clear all draw lines in selected range.

Values of *type*:

0	all lines in selected range
1	boarder line of selected range
2	left boarder line of cells in selected range
3	right boarder line of cells in selected range
4	top boarder line of cells in selected range
5	bottom boarder line of cells in selected range
6	slash line of cells in selected range
7	reverse slash line of cells in selected range

void AboutBox()

Return Value:

None.

Parameters:

None.

Remarks:

Display the version information of cell's Control Component.

void OnNewPageSetup(long pageno)

If add a new page, and it is the first time the page is set current page, trigger **OnNewPageSetup** event.

Parameters:

pageno page number

Return Value:

None.

void OnAllowMove(long oldcol, long oldrow, long newcol, long newrow, BOOL FAR* allow)

If drag and drop cells in cell using mouse, and change currently selected cell trigger **OnAllowMove** event.

Parameters:

<i>oldcol</i>	column number dragged where the cell located
<i>oldrow</i>	row number dragged where the cell located
<i>newcol</i>	column number dropped where the cell located
<i>newrow</i>	row number dropped where the cell located
<i>allow</i>	whether the destination cell is allowed to be set the data of the dragged cell.

Return Value:

None.

void OnAllowSizeRow(long row, BOOL FAR* allow)

If mouse arrow is focused on the bottom boarder of the row label cell, trigger **OnAllowSizeRow event**.

Parameters:

row row number where row label located

allow whether allow to change row height

Return Value:

None.

void OnAllowSizeCol(long col, BOOL FAR* allow)

If mouse arrow is focused on the right boarder of the column label cell, trigger **OnAllowSizeCol event**.

Parameters:

col column number where column label located

allow whether allow to change column width

Return Value:

None.

void OnAllowSizeTopLabel(BOOL FAR* allow)

If mouse arrow is focused on the bottom boarder of the corner cell, trigger **OnAllowSizeTopLabel** event.

Parameters:

allow whether allow change the height of column label

Return Value:

None.

void OnAllowSizeSideLabel(BOOL FAR* allow)

If mouse arrow is focused on the right boarder of the corner cell, trigger **OnAllowSizeSideLabel** event.

Parameters:

allow whether allow to change row label's width

Return Value:

None.

void OnColChange(long oldcol, long newcol)

If the column number of currently selected cell changes, trigger **OnColChange** event.

Parameters:

<i>oldcol</i>	column number of cell previously located
<i>newcol</i>	column number of cell currently located

Return Value:

None.

void OnRowChange(long oldrow, long newrow)

If row number of currently selected cell changes, trigger **OnRowChange** event.

Parameters:

<i>oldrow</i>	row number of cell previously located
<i>newrow</i>	row number of cell currently located

Return Value:

None.

void OnCellChange(long oldcol, long oldrow, long newcol, long newrow)

If change the currently selected cell, trigger **OnCellChange** event.

Parameters:

<i>oldcol</i>	column number of cell previously located
<i>oldrow</i>	row number of cell previously located
<i>newcol</i>	column number of cell currently located
<i>newrow</i>	row number of cell currently located

Return Value:

None.

void OnLeftColChange(long oldcol, long newcol)

If scroll horizontal scroll bar or change page, make the first column in current visible range change, trigger **OnLeftColChange** event.

Parameters:

<i>oldcol</i>	previous column number
<i>newcol</i>	current column number

Return Value:

None.

void OnTopRowChange(long oldrow, long newrow)

When scroll vertical scroll bar or change page, make the first row in current visible range change, trigger **OnTopRowChange** event.

Parameters:

<i>oldrow</i>	previous row number
<i>newrow</i>	current row number

Return Value:

None.

void OnPageChange(long oldpage, long newpage)

If change currently displayed page, trigger **OnPageChange** event.

Parameters:

oldpage previous page number

newpage current page number

Return Value:

None.

void OnAllowInputFormula(long col, long row, BOOL FAR* allow)

If try to edit formula in formula cell, trigger **OnAllowInputFormula** event.

Parameters:

<i>col</i>	column number of formula cell
<i>row</i>	row number of formula cell
<i>allow</i>	whether allow changing

Return Value:

None.

void OnAllowChartGuide(long col, long row, BOOL FAR* allow)

If try to edit chart in chart cell, trigger **OnAllowChartGuide** event.

Parameters:

<i>col</i>	column number of chart cell
<i>row</i>	row number of chart cell
<i>allow</i>	whether allow to change

Return Value:

None.

void OnLClickGrid(long col, long row, BOOL updn)

When click cell using mouse right key, trigger **OnLClickGrid** event.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>updn</i>	whether to hold down or release. TRUE means hold down, FALSE means release.

Return Value:

None.

void OnRClickGrid(long col, long row, BOOL updn)

When click cell using mouse right key, trigger **OnRClickGrid** event.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>updn</i>	whether to hold down or release. TRUE means hold down, FALSE means release.

Return Value:

None.

void OnDClickGrid(long col, long row)

When double click cell using mouse left key, trigger **OnDClickGrid** event.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located

Return Value:

None.

void OnLClickTopLabel(long col, BOOL updn)

When click cell using mouse left key, trigger **OnLClickTopLabel** event.

Parameters:

<i>col</i>	column number of column label cell
<i>updn</i>	whether to hold down or release. TRUE means hold down, FALSE means release.

Return Value:

None.

void OnRClickTopLabel(long col, BOOL updn)

When click column label cell using mouse right key, trigger **OnRClickTopLabel** event.

Parameters:

<i>col</i>	column number of column label cell
<i>updn</i>	whether to hold down or release. TRUE means hold down, FALSE means release.

Return Value:

None.

void OnDClickTopLabel(long col)

When double click column cell using mouse left key, trigger **OnDClickTopLabel** event.

Parameters:

col column number of column label cell

Return Value:

None.

void OnLClickSideLabel(long row, BOOL updn)

When click row label cell using mouse left key, trigger **OnLClickSideLabel** Event.

Parameters:

<i>row</i>	row number of row label cell
<i>updn</i>	whether to hold down or release. TRUE means hold down, FALSE means release.

Return Value:

None.

void OnRClickSideLabel(long row, BOOL updn)

When click row label cell using mouse right key, trigger **OnRClickSideLabel** Event.

Parameters:

<i>row</i>	row number of row label cell
<i>updn</i>	whether to hold down or release. TRUE means hold down, FALSE means release.

Return Value:

None.

void OnDClickSideLabel(long row)

When double click row label cell using mouse left key, trigger **OnDClickSideLabel** Event.

Parameters:

row row number of row label cell

Return Value:

None.

void OnLClickCorner(BOOL updn)

When click corner cell using mouse left key, trigger **OnLClickCorner** event

Parameters:

updn whether to hold down or release. **TRUE** means hold down, **FALSE** means release.

Return Value:

None.

void OnRClickCorner(BOOL updn)

When click corner cell using mouse right key, trigger **OnRClickCorner** event

Parameters:

updn whether to hold down or release. **TRUE** means hold down, **FALSE** means release.

Return Value:

None.

void OnDClickCorner()

When double click corner cell using mouse left key, trigger **OnDClickCorner** event

Parameters:

None.

Return Value:

None.

void OnCompareSortData(const VARIANT FAR& data1, const VARIANT FAR& data2)

When sorting the cells in one column, trigger **OnCompareSortData** event

Parameters:

<i>data1</i>	compared data
<i>data2</i>	comparing data
<i>compareResult</i>	result of comparement

Return Value:

None.

Using the event, user can program his own program to compare.

Result of comparement:

-1	<i>data1 < data2</i>
0	<i>data1 = data2</i>
1	<i>data1 > data2</i>

void OnKeyDown(long FAR* keycode)

When user hold down keyboard, trigger **OnKeyDown** event

Parameters:

keycode key code

Return Value:

None.

void OnCharDown(long FAR* charcode)

When user hold down character key, trigger **OnCharDown** event

Parameters:

charcode

ASCII code of the character

Return Value:

None.

void OnExecuteUserFunc(LPCTSTR name, short rettype, short par anum, long FAR* paratype, VARIANT FAR* funcResult)

When calculate user defined function, trigger **OnExecuteUserFunc** event

Parameters:

<i>name</i>	user defined function name
<i>rettype</i>	returned value type
<i>par anum</i>	number of parameter function needs
<i>paratype</i>	pointer to parameter type
<i>funcResult</i>	pointer to function result

Return Value:

None.

Remarks:

You must use DoFetchFuncParameter() method to get parameters' value, please see VCDEMOjE

BOOL DoGetColWidthVB(col As Long, width As Variant)

Return Value:

If there is column number, return **TRUE**, else return FALSE.

Parameters:

col column number

width width

Remarks:

Get the width of specified column.

BOOL DoGetRowHeightVB(row As Long, height As Variant)

Return Value:

If there is row number, return TRUE, else return FALSE

Parameters:

row row number

height height

Remarks:

Get the height of specified row.

BOOL DoGetJoinRangeVB(col As Long, row As Long, startcol As Variant, startrow As Variant, endcol As Variant, endrow As Variant)

Return Value:

If succeed, return **TRUE**, else return **FALSE**.

Parameters:

<i>col</i>	column number of cell
<i>row</i>	row number of cell
<i>startcol</i>	column number of start column
<i>startrow</i>	row number of start row
<i>endcol</i>	column number of end column
<i>endrow</i>	row number of end row

Remarks:

Get the position of specified combined cells.

BOOL DoAddUserFunctionVB(functype As String, name As String, rtype As Long, paranum As Long, paratype As String, funchelpstring As String)

Return Value:

If succeed, return TRUE.else return FALSE

Parameters:

<i>functype</i>	function type
<i>name</i>	function name
<i>rtype</i>	type of function result, valid value: 0, result is numeric£»1, result is a string£»4, result is string or numeric.
<i>paranum</i>	number of parameters
<i>paratype</i>	parameter type
<i>funchelpstring</i>	description of function

Remarks:

Add user defined function.

Parameter type :

"V"&"v"	numeric type
"S"&"s"	string type
"C"&"c"	cell type
"A"&"a"	area type
"D"&"d"	numeric or string type

BOOL DoCalculateExprVB(expr As String, rettype As Integer, exprresult As Variant)

Return Value:

If succeed, return **TRUE**, else return **FASLE**.

Parameters:

<i>expr</i>	expression
<i>rettype</i>	result type of expression
<i>exprresult</i>	expression result

Remarks:

Calculate the specified expression.

If result type is 1, return a string. If result type is 0, return numeric value.

BOOL DoGetFirstSelectedCellVB(col As Variant, row As Variant)

Return Value:

If succeed, return TRUE, else return FALSE

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

Remarks:

Get the first cell position in selected cells.

void DoGetNextSelectedCellVB(col As Variant, row As Variant)

Return Value:

If succeed, return TRUE, else return FALSE

Parameters:

col the number of the column where the cell located

row the number of the row where the cell located

Remarks:

Get the next cell's position in select cells.

void OnAllowChartGuideVBS(ByVal col As Long, ByVal row As Long, allow As Variant)

When editing the chart in chart cell, trigger **OnAllowChartGuideVBS** event

Parameters:

<i>col</i>	column number of chart cell
<i>row</i>	row number of chart cell
<i>allow</i>	whether allow changing. If it is 0, change is not allowed£»otherwise change is allowed.

Return Value:

None.

void OnAllowEditCell (long col, long row, BOOL FAR* allow)

When editing cell, trigger **OnAllowEditCell** event

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>allow</i>	whether allow changing. If it is 0, change is not allowed£»otherwise change is allowed.

Return Value:

None.

void OnAllowEditCellVBS (ByVal col As Long, ByVal row As Long, allow As Variant)

When editing cell, trigger **OnAllowEditCellVBS** event

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>allow</i>	whether allow changing. If it is 0, change is not allowed£»otherwise change is allowed.

Return Value:

None.

void OnAllowInputFormulaVBS(ByVal col As Long, ByVal row As Long, allow As Variant)

When editing the formula in formula cell, trigger **OnAllowInputFormulaVBS** event

Parameters:

<i>col</i>	column number of formula cell
<i>row</i>	row number of formula cell
<i>allow</i>	whether allow changing. If it is 0, change is not allowed£»otherwise change is allowed.

Return Value:

None.

void OnAllowMoveVBS(ByVal oldcol As Long, ByVal oldrow As Long, ByVal newcol As Long, ByVal newrow As Long, allow As Variant)

When drag-drop cell in a cell, if change currently selected cell, trigger **OnAllowMoveVBS** event

Parameters:

Oldcol column number of the dragged cell

oldrow row number of the dragged cell

newcol column number of the dropped cell

newrow row number of the dropped cell

allow Whether destination cell accept the data dropped. If it is 0, change is not allowed; else, change is allowed.

Return Value:

None.

void OnAllowDeleteCell(ByVal col As Long, ByVal row As Long, allow As Variant)

When user press down DEL, trigger **OnAllowDeleteCell** event

Parameters:

col, row cell to be deleted

allow whether allow deleting. If it is 0, deleting is not allowedE»else deleting is allowed.

Return Value:

None.

void OnAllowSizeRowInGrid(long col, long row, VARIANT FAR* allow)

When user set row height using mouse in a cell, trigger this event.

If `PropertyAllowSizeRowInGrid` is *true*, user can set row height in the cell using mouse. Unless use this event to disallow, user can set row height in scrolling range, but can not set row height in non-scrolling range.

Parameters:

col, row cell of the mouse focused on

allow whether allow. If it is 0, setting is not allowed; otherwise setting is allowed.

void OnAllowSizeColInGrid(long col, long row, VARIANT FAR* allow)

When user set column width using mouse in a cell, trigger this event.

If `PropertyAllowSizeColInGrid` is *true*, user can set column width in the cell using mouse. Unless use this event to disallow, user can set column width in scrolling range, but can not set column width in non-scrolling range.

Parameters:

col, row cell of the mouse focused on

allow whether allow. If it is 0, setting is not allowed; otherwise setting is allowed.

void OnAllowSizeColVBS(ByVal col As Long, allow As Variant)

When mouse arrow focus on the right boarder of the column label cell, trigger **OnAllowSizeColVBS** event

Parameters:

col column number of column label
allow whether allow changing column width. If it is 0, changing is not allowed£»else change is allowed.

Return Value:

None.

void OnAllowSizeRowVBS(ByVal row As Long, allow As Variant)

When mouse arrow focus on the bottom boarder of row label, trigger **OnAllowSizeRowVBS** event

Parameters:

<i>row</i>	row number of row label
<i>allow</i>	whether allow changing row height. If it is 0, changing is not allowedE»else change is allowed.

None.

void OnAllowSizeSideLabelVBS(allow As Variant)

When mouse arrow focus on the right boarder of corner cell, trigger **OnAllowSizeSideLabelVBS** event

Parameters:

allow whether allow changing column width of row label. If it is 0, changing is not allowed else change is allowed.

Return Value:

None.

void OnAllowSizeTopLabelVBS(allow As Variant)

When mouse arrow focus on the bottom boarder of corner cell, trigger **OnAllowSizeTopLabelVBS** event

Parameters:

allow whether allow changing row height of column label. If it is 0, changing is not allowedE»else change is allowed.

Return Value:

None.

void OnCharDownVBS(charcode As Variant)

When user press down character key, trigger **OnCharDownVBS** event

Parameters:

<i>charcode</i>	ASCII code of character
-----------------	-------------------------

Return Value:

None.

void OnCompareSortDataVBS(ByVal data1 As Variant, ByVal data2 As Variant, comparerresult As Variant)

When sorting the data in a column, trigger **OnCompareSortDataVBS** event

Parameters:

<i>data1</i>	compared data
<i>data2</i>	comparing data
<i>compareResult</i>	result of the compare

Return Value:

None.

Use this event, user can program their own comparing function.

CompareResult includes:

-1	<i>data1 < data2</i>
0	<i>data1 = data2</i>
1	<i>data1 > data2</i>

void OnExecuteUserFuncVBS(ByVal name As String, ByVal rttype As Long, ByVal paranum As Long, ByVal paratype As String, funcResult As Variant)

When calculate user defined function, trigger **OnExecuteUserFuncVBS** event

Parameters:

<i>name</i>	user defined function name
<i>rettype</i>	type of result
<i>paranum</i>	number of parameter function needed
<i>paratype</i>	pointer to the type of parameter
<i>funcResult</i>	pointer to the result of the function

Return Value:

None.

void OnUserFuncGuide(long parent, LPCTSTR funcname, VARIANT FAR* guidestring)

When user click the function guide button in inputting function(if current function is user defined function), trigger **OnUserFuncGuide** event

Parameters:

<i>parent</i>	handle of father window(HWND)
<i>funcname</i>	function name
<i>guidestring</i>	returned string when finished function guide

Return Value:

None.

void OnGetCellData(long col, long row, VARIANT FAR* data, BOOL FAR* changed)

When refresh cell, trigger **OnGetCellData** event

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>data</i>	data in cell
<i>changed</i>	whether to change the data

Return Value:

None.

void OnGetCellDataVBS(ByVal col As Long, ByVal row As Long, data As Variant, changed As Variant)

When refresh cell, trigger **OnGetCellDataVBS** event.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>data</i>	data in cell
<i>changed</i>	whether to change its data. If it is 0, there is no changeE»otherwise there is change.

Return Value:

None.

void OnKeyDownVBS(keycode As Variant)

When user press key, trigger **OnKeyDownVBS** event.

Parameters:

keycode code of the key

Return Value:

None.

void DoSetDropGridCell(long col, long row, long areacol1, long arearow1, long areacol2, long arearow2, long pageno, BOOL onlycol, long validcol, BOOL hidetag, BOOL readonly)

Set current cell the cell to be a drop grid cell.

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>areacol1, arearow1, areacol2, arearow2</i>	range of the data imported by drop grid
<i>pageno</i>	page number of the range
<i>onlycol</i>	whether there is only one column can be selected for the data in the drop grid
<i>validcol</i>	If onlycol is TRUE, specify data in which column can be selected, otherwise this parameter has no meaning.
etc: In area C3:G8, invalid column is E, so validcol = 4	
<i>hidetag</i>	whether hide drop down button
<i>readonly</i>	whether the cell is ediCell

void DoFind()

Return Value:

None.

Parameters:

None.

Remarks:

Display Search Dialog Box.

void DoReplace()

Return Value:

None.

Parameters:

None.

Remarks:

Display Search Dialog Box.

void DoLocate()

Return Value:

None.

Parameters:

None.

Remarks:

Display Locate Dialog Box.

void DoShowFindDialog(BOOL show)

Return Value:

None.

Parameters:

show: Set the property of Search Replace Dialog Box visible or invisible.

Remarks:

Set the property of Search Replace Dialog Box visible or invisible.

void DoCloseFindDialog()

Return Value:

None.

Parameters:

None.

Remarks:

Close Search Replace Dialog Box.

BOOL DoOpenFileDbase(short dbasetype, LPCTSTR filename, VARIANT FAR* Cells)

Open file type database.

Parameters:

dbasetype type of file type database

dbasetype includes:

- 0 "Access file (*.mdb)",
- 1 "dBASE III file (*.dbf)",
- 2 "dBASE IV file (*.dbf)",
- 3 "dBASE 5 file (*.dbf)",
- 4 "Paradox 3.x file (*.db)",
- 5 "Paradox 4.x file (*.db)",
- 6 "Paradox 5.x file (*.db)",
- 7 "Btrieve file (*.ddf)",
- 8 "FoxPro 2.0 file (*.dbf)",
- 9 "FoxPro 2.5 file (*.dbf)",
- 10 "FoxPro 2.6 file (*.dbf)",
- 11 "Lotus WKS file (*.wks)",
- 12 "Lotus WK1 file (*.wk1)",
- 13 "Lotus WK3 file (*.wk3)",
- 14 "Lotus WK4 file (*.wk4)",
- 15 "Text file (*.txt)",
- 17 "Excel 3.0 file (*.xls)",
- 18 "Excel 4.0 file (*.xls)",
- 19 "Excel 5.0 file (*.xls)",
- 20 "Excel 7.0 file (*.xls)",
- 21 "Excel 97 file (*.xls)"

filename database file name, for example: C:\Documnet\orders.mdb

Cells If the function returns TRUE, this parameter returns all the cell name string in database. Each cell name in the database is separated by '\x0d', '\x0a'.

Return Value

TRUE open database successfully

FALSE fail to open. The most possible reason is there is no ODBC driver installed .

BOOL DoOpenFileDbaseVfp(short dbasetype, LPCTSTR filename)

Open database of file type. Such function is used in Visual Foxpro.

Parameters:

dbasetype file type database

dbasetype including values:


```
0 "Access file(*.mdb)",
1 "dBASE III file(*.dbf)",
2 "dBASE IV file(*.dbf)",
3 "dBASE 5 file(*.dbf)",
4 "Paradox 3.x file(*.db)",
5 "Paradox 4.x file(*.db)",
6 "Paradox 5.x file(*.db)",
7 "Btrieve file(*.ddf)",
8 "FoxPro 2.0 file(*.dbf)",
9 "FoxPro 2.5 file(*.dbf)",
10 "FoxPro 2.6 file(*.dbf)",
11 "Lotus WKS file(*.wks)",
12 "Lotus WK1 file(*.wk1)",
13 "Lotus WK3 file(*.wk3)",
14 "Lotus WK4 file(*.wk4)",
15 "Text file(*.txt)",
17 "Excel 3.0 file(*.xls)",
18 "Excel 4.0 file(*.xls)",
19 "Excel 5.0 file(*.xls)",
20 "Excel 7.0 file(*.xls)",
21 "Excel 97 file(*.xls)"
```

filename database file name, for example: C:\Documnet\orders.mdb

Return Value

TRUE Open database successfully

FALSE fail to open.The most possible reason is there is no ODBC driver installed .

BOOL DoCreateODBCDSN(LPCTSTR drivename, LPCTSTR description)

Create ODBC data source.

Parameters:

drivename name of driver

description description of data source, each parameter is separated by semicolon (semicolon is not used except this meaning)

Example: Visual FoxPro data source:

"Microsoft Visual FoxPro Driver",

"DSN=MYDATA;DESCRIPTION=TEST FOXPRO;SourceType=DBC;SourceDB=F:\t\DATA\books.dbc;"

BOOL DoRemoveODBCDSN(LPCTSTR dsnname)

Delete ODBC data source.

Parameters:

drivername ODBC data source name

BOOL DoOpenODBCDbase(LPCTSTR connectstr, VARIANT FAR* Cells)

Open ODBC data source.

Parameters:

Connectstr ODBC connection string, for example: "DSN=mydataSource;UID=SA;PWD=abc123" is a demo connection string, mydataSource is the data source name created in ODBC manager, SA is the user ID of the data source, abc123 is password of the ID.

Note: If this parameter is NULL or a null string, ODBC Data Source Dialog Box is displayed, and data source can be set in Dialog Box.

Cells If function return TRUE, this parameter return all cell name strings separated by character return '\x0d', '\x0a'.

Return Value

TRUE Open database successfully

FALSE fail to open. The most possible reason is there is no ODBC driver installed .

BOOL DoOpenODBCDbaseVfp(LPCTSTR connectstr)

Open ODBC data source. This function is used in Visual Foxpro.

Parameters:

connectstr string to connect ODBC, for example "DSN=mydataSource;UID=SA;PWD=abc123" is a typical connection string. mydataSource is the data source name created in ODBC manager, SA is the data source user's ID, abc123 is the ID's password.

Note: If this parameter is NULL or a null string, ODBC Data Source Dialog Box is displayed, and data source can be set in Dialog Box.

Return Value

TRUE Open database successfully

FALSE fail to open. The most possible reason is there is no ODBC driver installed .

BOOL DoDumpDbaseData(LPCTSTR sqlstring, long col, long row, long page, BOOL withfieldname, VARIANT FAR* dumpcols, VARIANT FAR* dumprows)

Dump the finding results to page. Note, the page must exist.

Parameters:

sqlstring SELECT sentence, for example "SELECT * from Cell1 where id > 100 ". Please refer to SQL user guide for its syntax.

col, row start position of the data saved on the cell

withfieldname TRUE shows field name, FALSE does not show

page page number Note: The page number must exist.

dumpcols, dumprows These two parameters are returned by this function, indicate how many rows and columns data are dumped to cell.

Return Value

TRUE Success. The size of page will be reset to comply with the finding result.

FALSE Failure.

Note: Database must be open before call this function.

BOOL DoDumpDbaseDataVfp(LPCTSTR sqlstring, long col, long row, long page, BOOL withfieldname)

Dump the finding results to page. Note, the page must exist. This function is used in Visual Foxpro.

Parameters:

sqlstring SELECT sentence, for example "SELECT * from Cell1 where id > 100 ". Please refer to SQL user guide for its syntax.

col, row start position of the data saved on the cell

withfieldname TRUE shows field name, FALSE does not show

page page number. Note: The page number must exist.

dumpcols, dumprows These two parameters are returned by this function, indicate how many rows and columns data are dumped to cell.

Return Value

TRUE Success. The size of page will be reset to comply with the finding result.

FALSE Failure.

Note: Database must be open before call this function.

BOOL DoDumpDbaseCell(LPCTSTR Cellname, long col, long row, long page, BOOL withfieldname, VARIANT FAR* dumpcols, VARIANT FAR* dumprows)

Dump the contents of one cell in database to a page. Note, the page must exist.

Parameters:

Cellname cell name of database, can be found from the function to open database.
col, row start position of the data saved on the cell
withfieldname TRUE shows field name, FALSE does not show
page page number. Note: The page number must exist.
dumpcols, dumprows These two parameters are returned by this function, indicate how many rows and columns data are dumped to cell.

Return Value

TRUE Success. The size of page will be reset to comply with the finding result.
FALSE Failure.

Note: Database must be open before call this function.

void DoCloseDbase()

Close database

Caution: This function will be called automatically when Control Component object deconstructs.

BOOL DoSaveFile(LPCTSTR filename)

Save contents of cell to a file. The file has the same format with cell file.

Parameters:

filename file name

Return Value

- >0 success (Note: It is not more than or equal)
- 1 can not find the accessory of cell Control Module.
- 2 not have enough disk space;
- 3 with no operation right;
- 4 hardware failure
- 5 share failure
- 6 invalid file name or invalid path
- 7 can not open file created by new version cell.
- 8 invalid file type
- 99 unknown error

long DoOpenFile(LPCTSTR filename)

Open file, and display contents in page. Can open files created by cell.

Parameters:

filename file name

Return Value

- >0 success (Note not more than or equal)
- 1 can not find the accessory of cell Control Module.
- 2 not have enough disk space;
- 3 with no operation right;
- 4 hardware failure
- 5 share failure
- 6 invalid file name or invalid path
- 7 can not open file created by new version cell.
- 8 invalid file type
- 99 unknown error

**void DoSetButtonCellEx(long col, long row, LPCSTR text,
LPCTSTR runstring)**

Set a cell to be a button cell

Parameters:

col the column number of the cell

row the row number of the cell

text text on the button

runstring the command to be run(exe file name or registered document)

Return Value

none

void DoCalculatePage(long page)

Calculate current page's formula

Parameters:

page Page number(from 0)

Return Value

None

void DoUndo()

Cancel current operation

Only user's operation can be cancelled.

void DoRedo()

Redo the operation currently cancelled.

void DoDiscardUndo()

Discard Undo information, disallow cancel

void DoGetUndoState()

Get Undo information

Return Value

TRUE	currently it can Undo
FALSE	currently it can not Undo

void DoGetRedoState()

Get Redo information

Return Value

TRUE	currently it can Redo
FALSE	currently it can not Redo

void DoSetMessageTitle(VARIANT FAR& title)

Set message box's title of Control Module (the default is cell)

Return Value

title the string of the message box's title

void DoSetCellReadOnly(long col, long row, BOOL readonly)

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>readonly</i>	whether set Readonly property, TRUE set , FALSE clear it

**void DoSetPageCellData(long col, long row, long page, const
VARIANT FAR& data)**

Return Value:

None.

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>page</i>	the number of the page where the cell located(from 0)
<i>data</i>	data to be set

Remarks:

Set data to cell.

When *col* or *row* is -1, the cell set is column label or row label. When they are both -1, the cell set is corner cell.

BOOL DoGetPageCellData(long col, long row, long page, VARIANT FAR* data)

Return Value:

If succeed return **TRUE**, otherwise **FALSE**.

Parameters:

<i>col</i>	the column number of where the cell located
<i>row</i>	the row number of where the cell located
<i>page</i>	the page number of where the cell located
<i>data</i>	return data's pointer

Remarks:

Get specified cell's data.

void OnEditError(ByVal col As Long, ByVal row As Long, text As Variant)

When the input value of edited cell is wrong, trigger its event

Parameters:

<i>col</i>	column number of the cell
<i>row</i>	row number of the cell
<i>text</i>	string user written

Return Value:

None.

void OnEditFinish(long col, long row, LPCTSTR string, VARIANT FAR* valid)

This event occurs when edit finished

Parameters:

<i>col</i>	col number
<i>row</i>	row number
string	input string
valid	valid or not(return to CELL, won't input when this value set to 0)

Return Value:

none.

void OnAllowMoveToPage(ByVal pageno, allow As Variant)

When user change current page, it will trigger the page's event

Parameters:

pageno new page number

allow When *allow* is set FALSE, user will not change to page *pageno*

Return Value:

none.

void DoSetCellPrintable(long col, long row, long size, BOOL hide)

Set whether to hide the contents of cell when printing

Return Value:

None

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located
<i>hide</i>	TRUE indicate hiding cell's contents

void DoResetContent()

Return Value:

None.

Remarks:

Clear all pages' contents—only one vacant page left.

void OnCellButtonClicked(long col, long row)

This event occurs when user click the button in a cell

Parameters:

<i>col</i>	the number of the column where the cell located
<i>row</i>	the number of the row where the cell located

Return Value:

None.

void OnPrint(VARIANT FAR* processed)

Occur when user click the print button in print preview

Parameters:

processed Indicate processed the event or not, if processed set to TRUE, CELL won't send data to printer.

Return Value:

None.

Note: CELL print the same content as what you see in print preview, but you can change this with this event. You can print preview page 2's content, but send page 2 to printer (use DoPrint) with this event.

BOOL DoAppendPageFromFile(LPCTSTR filename, long startpage, long count)

append page from other file

Parameters:

filename the source file name
startpage append start from filename file's startpagepage
count pages number want to append

Return Value:

TRUE OK
FALSE Fail

BOOL IsModified()

the modified state of the contentin control

Return Value:

TRUE modified

FALSE not modified

Note:

Set modified flag (DoSetModifiedFlag()) to FALSE after initialize Control, test it's value when exit.

void DoSetModifiedFlag(BOOL modified)

set modified state

Parameters:

modified the flag value to set

Return Value:

none

BOOL DoCopyPage(long dest, long source)

copy page(dest, source should exist)

Parameters:

dest dest page number
source source page number

Return Value:

TRUE OK
FALSE Fail

void DoPrintPage(long page, BOOL showdlg)

print page

Return Value:

none

Parameters:

page page number

showdlg show dialog or not

void DoPrintPreviewPage(long page, BOOL allowPageset)

print previewpage

Return Value:

none

Parameters:

page page number

allowPageset allow set page pprint parameter

**void DoSetHintButtonCell(long col, long row, LPCTSTR text,
LPCTSTR mesg, BOOL hide, BOOL canEdit)**

set hint style button cell

Parameters:

<i>col</i>	col number
<i>row</i>	row number
<i>text</i>	button text
<i>mesg</i>	pop up message, if empty, will not pop message
hide	hide if focus lost
canEdit	allow input in this cell

Return Value:

none

long DoSaveTextFile(LPCTSTR filename, long format)

Output current page's content to a text file

Parameters:

filename: text file name

format: text file format, valid value:

- 0 separate by tab
- 1 separate by 2 continuous space chars
- 2 separate by comma

Return Value:

- 1 success
- 0 parameter invalid
- 1 fail, can't create file or path invalid

long DoSaveHtmlFile(LPCTSTR filename)

Output put current page's content to HTML file.

Parameters:

filename: HTML file name

Return Value:

1 success

-1 fail, can't create file or path invalid

long DoReadTextFile(LPCTSTR filename, long format)

Read text file's content to current page and resize page.

Parameters:

filename: text file name

format: text file format, valid value:

- 0 separate by tab
- 1 separate by at least 2 continuous space chars or tab
- 2 separate by comma

Return Value:

- 1 success
- 0 parameter invalid
- 1 fail, file can't open or path invalid

BOOL GetUserDouble(LPCTSTR filename, short index, double FAR* value)

There is a buffer in cll file for user to save extra parameters(10 numeric value and 10 string), use this method to get the numeric value.

Parameters:

filename file name, if it's NULL or empty string, indicate the user variable in current Control.

index numeric variable index, valid value 0 – 9

value numeric value to return

Return Value:

TRUE success

FALSE open file failed

Note: It's slower to access data from disk file.

BOOL GetUserString(LPCTSTR filename, short index, VARIANT FAR* value)

There is a buffer in cll file for user to save extra parameters(10 numeric value and 10 string), use this method to get the string.

Parameters:

filename file name, if it's NULL or empty string, indicate the user variable in current Control.

index string variable index, valid value 0 – 9

value string to return

Return Value:

TRUE success

FALSE open file failed

Note: It's slower to access data from disk file.

BOOL SetUserDouble(LPCTSTR filename, short index, double value)

There is a buffer in dll file for user to save extra parameters(10 numeric value and 10 string), use this method to set the numeric value.

Parameters:

filename file name, if it's NULL or empty string, indicate the user variable in current Control.

index numeric variable index, valid value 0 – 9

value numeric value to set

Return Value:

TRUE success

FALSE open file failed

Note: It's slower to access data from disk file.

BOOL SetUserString(LPCTSTR filename, short index, LPCSTR value)

There is a buffer in cll file for user to save extra parameters(10 numeric value and 10 string), use this method to set the string.

Parameters:

filename file name, if it's NULL or empty string, indicate the user variable in current Control.

index string variable index, valid value 0 – 9

value string to return

Return Value:

TRUE success

FALSE open file failed

Note: It's slower to access data from disk file.

