CellType

The type of cell.

Usage

```
Set_Property(OLECtrlEntID, "OLE.CellType[field; record]", Array)
```

Values

Array has the following structure:

Pos	Name	Type	Description	Default
<1>	Type	Option	The cell's type. (See Remarks)	Text
<2>	Parameters	Multivalue	Parameters specific to the Type, (See Remark)	N/A
<3>	Button Background	Color Fill	The button background	None
<4>	Button Background Hot	Color Fill	The button background when the user hovers over it	None
<5>	Button Background Pushed	Color Fill	The button background when the user presses it	None
<6>	Button Background Disabled	Color Fill	The button background when disabled. (4.0.1 and later only)	None
<7>	Button Foreground Disabled	Color	The button text color when disabled. (4.0.1 and later only)	None

Indices

Index	Description
field	Index to an existing field
record	Index to an existing record

Remarks

The CellType property establishes a cell's generic behavior. By default, all cells are "Text" cells, having containg only plain text without any kind of validation. Always check this property's documentation because future update may include new cell types.

The possible values for the Type field are:

Value	Abbr.	Description
Text	TXT	A normal cell containing unvalidated and unconverted text
Option	OPT	A text cell with an option button that fires OnOptionClick events
Hyperlink	HYP	A cell whose text is clickable and fires OnHyperClick events
Check Box	CHB	A cell with a check box that fires OnCheckChanged events
Push Button	PUB	A cell whose entire area is that of a standard push button
Flashing	FLS	A text cell whose text flashes to alert the user
Combo	COB	A text cell with an option button that shows a dropdown list from which the user can select a new value for the cell

Many cell types require extra data for initialization. The extra data is provided via the Parameters field of this property (field 2).

You must always set the first field of this property, even if you just want to update the type parameter information. For example, to change an option button image from from 1 to 0, you must pass "Option":@FM:

Text

The Text cell type is simply a cell with no bells and whistles. It only contains and manipulates text, and all cells are Text cells be default. This type takes no parameters and ignores the Parameter field.

Option

Pos	Name	Type	Description
<2, 1>	Content	Text	If this is a number, it is an index to the image to appear on the option button face. Otherwise, this is the text to appear on the button. **(Added Text in 4.1.20)**
<2, 2>	Font	Font	The font used to render the button text. **(Added 4.1.20)**
<2, 3>	Color	Color	The color of the button text. **(Added 4.1.20)**

The Option cell type adds an option button on the right side of the cell, which will fire OnOptionClick events whenever clicked. Otherwise, the cell acts just like a Text cell.

The **Content** parameter sets the content to appear on the face of the button. If you pass a number, then you are setting it to an index pointing to an image frame in the image specified by the ImageList property. If you pass anything else, then you are setting text to appear on the button face. To specify the default drop down arrow image, leave Content blank.

The **Font** parameter sets the font used to render text. The default is Tahoma 7.5pt.

The **Color** parameter sets the color used to render text.

Prior to **4.1.20**, you could only set <2, 1> to an image index.

Hyperlink

Pos	Name	Type	Description
<2, 1>	HyperFont	Font	The font when the text is hot
<2, 2>	HyperColor	Color	The color when the text is hot
<2, 3>	As Data	Boolean	A flag indicating whether or not the contents of the cell are treated as data
<2, 4>	Editable	Boolean	A flag indicating whether or not the user can edit the cell

A Hyperlink cell is one whose text can be clicked, which will fire OnHyperClick events. You can use the event to link the text to other forms, data, web pages, etc.

The **HyperFont** parameter is the Font to be used when the text is hot. A cell's text is hot when the mouse hovers over it. The HyperColor parameter is the Color to be used when the text is hot. When the text is not hot, then the cell's CellColors values are used.

The **As Data** parameter determines if the contents of a hyperlink cell are to be treated as data. If so, which is the default, then the contents can be manipulated using the LIST and ARRAY properties. If not, then ARRAY and LIST will skip the hyperlink cell altogether. The **Editable** parameter determines if the user may edit the contents of a hyperlink cell. When turned on, the user must double-click within the cell but outside of the hyperlink to enter into edit mode (or use accelerators).

Check Box

Pos	Name	Type	Description
<2, 1>	IsFlat	Boolean	Determines if the check box is flat or 3D
<2, 2>	AsData	Boolean	Determines if the check box should be treated as data
<2, 3>	ButtonDown	Boolean	Determines if the check box state changes when the button goes down instead of waiting for a full click. **(Added in 4.1.20)**

A Check Box cell contains a check box to the left of the cell's contents. The cell can still be edited if CellProtection allows for it. Thus, the embedded check box has it's own protection level accessible via the CellCheckEnabled property. The current value of a cell's check box is explicitly available via the CellCheck property.

The IsFlat parameter is a true/false value indicating whether the appearance of the check box is flat (true) or 3-dimensional (false). The **AsData** parameter is also a true/false value indicating whether the check box should be interpreted as the cell's data. When set to true, the ARRAY and LIST properties as well as the INSERT method modify the check box value rather than the cell's text.

The ButtonDown parameter affects the clicking behavior of the check box. Normally, to change the state of the check box, the user must press the left mouse button and release it within the bounds of the check box. Releasing the button outside the check box normally cancels the click. If you want your check box to be more responsive, set this value to 1. Doing so will instantly change the state of the check box as soon as the button is pressed.

Push Button

Pos	Name	Type	Description
<2, 1>	RowCopy	Boolean	If true, then buttons on new records copy the text and image of the button in the record above it
<2, 2>	EnterKey	Boolean	If true, then pressing the ENTER key clicks the button instead of navigating to the next cell
<2, 3>	As Data	Boolean	A flag indicating whether or not the contents of the cell are treated as data

The Push Button cell type converts the cell entirely into a button. This differs from the Option cell type in that an Option cell has a small button on the left side of the cell. You set a push button's caption and/or image using the cell's CellText and/or CellImage properties respectively. Alternatively, you may use the INSERT, ARRAY, and LIST properties to set button caption's as well.

The **RowCopy** parameter is a true/false value indicating whether or not buttons that appear on new rows duplicate the text and image of the corresponding button in the row above it. If false, then the developer is responsible for setting the button's text and/or image. The **EnterKey** parameter is also a true/false value indicating whether or not the user can "click" the button by pressing the ENTER key. When false, which is the default, the ENTER simply navigates to the next cell. In either case, the user can always click the button using the mouse or SPACE key.

When a button cell is clicked, the OnButtonClick event is fired. Capture the event to respond to user button clicks.

The **As Data** parameter determines if the contents of a push button cell are to be treated as data. If so, which is the default, then the contents can be manipulated using the LIST and ARRAY properties. If not, then ARRAY and LIST will skip the push button cell altogether.

Flashing

Pos	Name	Type	Description
<2, 1>	FlashColor	Color	The flashing color of the text
<2, 2>	Delay	Integer	The time in milliseconds to delay flashing
<2, 3>	Smooth	Boolean	If true, the text fades in and out of its flash color; if false, it simply alternates flash and text colors

The Flashing cell type can be used to alert the user to important data or changes in data. A flashing cell alternates between its foreground color and the provided **FlashColor** parameter. The flash frequency is determined by the **Delay** parameter, which takes a number representing milliseconds. Finally, the **S mooth**parameter determines if the text fades to and from the flash color or if it simply toggles directly between the two. Experiment with different settings to get the desired effect.

Flashing cells are exactly like text cells. They can be edited, protected, etc.

Combo

Pos	Name	Type	Description
<2, 1>	Columns List	Multivalue	@TM delimited list of column headings
<2, 2>	Column Settings List	Multivalue	@TM delimited list of column settings
<2, 3>	Data	Multivalue	Combo box data in ARRAY format; @TM delimited columns of data with @STM delimited rows
<2, 4>	Master Column	Integer	The column containing the return values
<2, 5>	Autofill	Boolean	If true, then autofill is enabled
<2, 6>	Case Sensitive Autofill	Boolean	If true, then autofill is case sensitive
<2, 7>	Number of Visible Rows	Integer	The number of viewable rows at a time
<2, 8>	Fire OnOptionClick	Boolean	If true, fire the OnOptionClick when user presses dropdown button
<2, 9>	Filtered Autofill	Boolean	If true, reduce the list to only those items that partially match the current contents
<2, 10>	Show Dropdown Immediately	Boolean	If true, shows the dropdown as soon as user enters edit mode
<2, 11>	LIST Format	Boolean	If true, then the data is in LIST format and not ARRAY format
<2, 12>	Autofill Column(s)	Integer	The column(s) against which autofill is performed
<2, 13>	Hide When Empty	Boolean	If true, then the dropdown hides when the user clears the cell
<2, 14>	Normal Navigation	Boolean	If true, then the up/down/pgup/pgdown keys navigate to other cells when the drop down is hidden
<2, 15>	Show Only When Empty	Boolean	If true and Show Immediately is true, then the dropdown only shows immediately when the cell's contents are empty
<2, 16>	No Show Autofill	Boolean	If true, then the drop down does not appear during autofill

<2, 17>	Clear Selection on Backspace or Delete	Boolean	If true, then the currently selected drop down item will be deselected when the user presses backspace or delete
<2, 18>	Show Popup While Read Only	Boolean	If true, then the user can click the drop down button and select an item but still cannot type in the cell
<2, 19>	Show Popup While Navigating	Boolean	If true and Normal Navigation is false, then pressing up/down/pgup/pgdown shows the popup
<2, 20>	Always Tab Out on Enter	Boolean	If true, then pressing Enter will tab out of the cell even if the drop down was visible
<2, 21>	Dropdown Threshold	Integer	The number of rows, at minimum, that must be visible within the dropdown for it to appear below the cell
<2, 22>	Dropdown Font	Font	The font used to render the drop down items
<2, 23>	Limit to List	Boolean	Determines if the cell's contents must be limited to the values in the dropdown.
<2, 24>	Match Anywhere	Boolean	Determines if autofill will also match what you type to any substrings within text, instead of just at the beginning of it.
<2, 25>	Row Height	Integer	The height of each row in the dropdown
<2, 26>	Force Cell Update	Boolean	Determines if the cell should be updated immediately after the user clicks an item in the dropdown **(Added in 4.1.20)**

The Combo cell type can be used for quickly selecting from a list of values. This style will show a dropdown list when the user edits the cell or presses the drop down button. Set the **Columns List** parameter if you want headers above your columns or omit it to have no headers.

The **Column Settings List** parameter lets you specify column alignment and width; it's an @TM delimited list of these values, and each element of the list has the structure Alignment:@STM:Width. The alignment sets the column's data justification and can be "Left", "Center", or "Right". You can optionally use "L", "C", and "R" respectively. The width can be used to limit the width of a column (if you expect it to contain long strings of data). If you want your columns to be autowidth, omit the Fixed Width. If you want your columns to dynamically resize during filtered autofill, set the width to "Dynamic" or "DYN".

The **Data** parameter is the most important as it establishes the dropdown list data. It is just like the edittable's ARRAY property except it uses @TM instead of @FM and @STM instead of @VM. There is currently no support for sorting, so you'll have to sort this array before you set it.

The **Master Column** parameter lets you specify which column contains the values that will be placed in the cell when the user selects it. This is also the column that the auto fill uses by default (see Autofill Column below). To use auto fill, set the **AutoFill** to 1 (although this is 1 by default). Auto fill is case insensitive by default, so set the **Case Sensitive Autofill** to 1 if you need it.

If you suspect that the combo box will have many rows of data, you may use the **Number of Visible Rows** parameter to limit the height of the combo box dropdown list. By default, the list fits as many rows as possible without leaving the screen. This parameter however, will limit the maximum height and display a scroll bar for the user.

In some cases, you may want the combo box dropdown to appear during autofill, but you'd rather show a popup window if the user presses the dropdown button. For such cases, set the **Fire OnOptionClick** parameter to 1 which will fire the OnOptionClick event instead of showing the combo box dropdown list.

The ninth parameter, **Filtered Autofill**, simply enhances the autofill technology by completely hiding items that do not partially match the current autofill, thus making the list smaller and smaller as the user narrows down the search. This conveniently allows the user to more easily select the closest item, especially if your list is not sorted. When this is turned off, autofill still works, but the list never changes its size. If this is turned on and Autofill is off, then the list still filters down as the user types, but no auto selection or filling occurs.

By default, the dropdown appears immediately when the user enters edit mode. If, however, you only want the dropdown to appear once the user starts typing, then set the **Show Immediately** parameter to False. This way, the user only sees the drop down during autofill. If both the Autofill parameter and this one are set to false, then the user must press the arrow keys to make the dropdown appear.

The **LIST Format** parameter is a flag allowing you to change how the data array is interpreted. Normally, the third parameter (Data) must be in ARRAY format (@TM delimited columns with @STM delimited rows). By setting this parameter to 1 however, you can pass the Data in LIST Format (@TM delimited rows and @STM delimited columns). This flags also affects the CellComboData property. Thus, once you choose the LIST Format, all data interaction is in LIST format.

The **Autofill Column(s)** parameter allows you to specify which column(s) the autofill is performed against. If omitted, then the autofill column is the same as the Master Column. Set this to one or more columns if you want to allow the user to autofill against something other than the Master Column. This list can be comma delimited or @SVM delimited. The order of this list also determines the order that will columns will be searched during autofill.

The next parameter, **Hide When Empty**, determines the behavior of the dropdown when the user clears the cell. By default, once the dropdown is visible it remains so until the user leaves the cell or makes a selection. If you'd rather that the dropdown hides when the user clears the cell, then set this parameter to true. This only has an effect if autofill with filtering is enabled.

The **Normal Navigation** parameter determines how to respond to navigation keys. By default, the up, down, pgup, and pgdown keys make the drop down appear and allow you to navigate the drop down. If you set this parameter to true, then these keys will only navigate the drop down when it is already visible. Otherwise, they move to other cells in the edit table.

The **Show Only When Empty** determines when the drop down appears and is only useful when the Show Immediately parameter is true. By default, the drop down will show immediately whenever the cell enters edit mode. If you set this parameter to true, then the drop down will show immediately only if the cell is empty. This is useful if you want to prompt users for new values.

The **No Show Autofill** determines whether or not the drop down appears during autofill. Normally, the drop down appears as the user starts to type so they can see the nearest match. If you set this parameter to true, then the drop down will not appear during autofill; the user will have to explicitely show the drop down if they want to see their nearest matches.

The **Clear Selection on Backspace or Delete** flag determines whether or not the currently selected drop down will be deselected when the user presses the backspace or delete key. This is useful when you need to allow for the possibility that an autofill item might match a master column value. This is usually only needed when the autofill column is different from the master column.

The **Show Popup While Read Only** flag, when set, allows the user to click the drop down button to show the combo drop down even though the cell is read only. The user can select an item in the list but will be unable to type within the cell to enter custom data. This is useful if you want to limit the user to the combo list.

The **Show Popup When Navigating** flag, when set to true, causes the combo drop down to appear when the user presses the up, down, pageup, or pagedown keys. This flag only works when Normal Navigation is off.

The **Always Tab Out on Enter** flag, when set to true, will move the user to the next cell when pressing Enter, even if the drop down was visible. When set to false, pressing Enter while the drop down is visible merely selects the item and keeps focus on the current cell, a behavior that better mimicks standard combo box controls. The flag is on by default.

The **Dropdown Threshold** value allows you to customize the behavior of the dropdown as it approaches the bottom of the screen. The dropdown has the effect of being "squished" between the cell and screen bottom until fewer rows than the threshold are visible, in which case the dropdown is moved above the cell. By default, the threshold is 3 rows. You can increase or decrease this value as desired. A value of 0 means that the dropdown will not appear above the cell until no rows are visible. A value of -1 causes the dropdown to appear above the cell as soon as it touches the bottom. In other words, -1 disables the "squish" effect.

The **Dropdown Font** value determines the font used to render the items in the dropdown list. It does not affect the font used to render the actual edittable cell.

The **Limit to List** value determines if the user is prevented from typing anything that does not appear in the dropdown list.

The **Match Anywhere** value modifies autofill behavior slightly. Normally, as the user types, the typed text is matched against all autofill columns (See value <2, 12> above) to find any strings that begin with the same text. When this is set to 1, however, the typed text is allowed to be found anywhere within a string. When this is enabled, note that autofill will only help the user find matching rows but will not autofill the text in the cell unless the match is at the beginning of a string.

The **Row Height** value customizes the height of each row in the combo dropdown.

The **Force Cell Update** value determines if the table should immediately force the cell to update when the user clicks an item in the combo box dropdown. This feature is only available in version 4.1.20 or later.

Button Backgrounds

Fields <3> - <7> establish custom colors for Option Buttons, Combo Box buttons, and Push Buttons. Therefore, they only apply to cell's of type "Option", "Combo", and "Push Button". The backgrounds can be set to any valid Color Fill value. The foreground colors can be set to any valid Color value.

These fields are "None" by default, which indicates that the table should render them using the standard sytem themes. These fields have been added primarily to allow you to keep your cell buttons consistent with your header cell's.

Example

```
////////////////////////////////
// TEXT

// Set a text cell
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Text")

////////////////////////////////
// OPTION

// Set an option cell using the default drop-down image
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Option")

// Set an option cell using an image from the ImageList property
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Option":@FM:1)

////////////////////////////////
// HYPERLINK

// Set a hyperlink cell
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Hyperlink")

// Set a hyperlink cell with custom font
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Hyperlink":@FM:"Ariel":@SVM:"10")
```

```
// Set a hyperlink cell with custom font and color
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Hyperlink":@FM:"Ariel":@SVM:"10":@VM:"{255, 0,
255}")

////////////////////////////////
// CHECKBOX

// Set a checkbox cell
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Check Box")

// Set a flat checkbox cell
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Check Box":@FM:1)

// Set a normal checkbox cell treating the check box state as data
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Check Box":@FM:0:@VM:1)

////////////////////////////////
// PUSH BUTTON

// Set a push button cell with text and image
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Push Button")
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellText[1; 1]", "Push Me!")
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellImage[1; 1]", 1)

// Set a push button cell whose text/image are NOT copied in new records
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Push Button":@FM:0)

// Set a push button cell that can be pressed by hitting the ENTER key
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Push Button":@FM:1:@VM:1)

////////////////////////////////
// FLASHING

// Set a flashing text cell
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Flashing")

// Set a flashing text cell with custom color
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Flashing":@FM:"{255, 0, 255}")

// Set a flashing text cell with custom color and delay
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Flashing":@FM:"{255, 0, 255}":@VM:1000)

// Set a flashing text cell with custom color, delay, and with smoothing effect
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; 1]", "Flashing":@FM:"{255, 0, 255}":@VM:1000:@VM:1)

////////////////////////////////
// COMBO

FirstNames = "Don"  :@STM:"Paul"    :@STM:"Frank":@STM:"Bob"      :@STM:"Kevin"
LastNames  = "Bakke":@STM:"Simonsen":@STM:"Tomeo":@STM:"Fernandes":@STM:"Fournier"
TypeData = ""
TypeData<1>    = "Combo"
TypeData<2, 1> = "First Name":@TM:"Last Name"
TypeData<2, 2> = "L":@STM:100:@TM:"L"
TypeData<2, 3> = FirstNames:@TM:LastNames
TypeData<2, 4> = 2                          ;// column 2 contains the values we care about
TypeData<2, 5> = 1                          ;// auto fill on
TypeData<2, 6> = 0                          ;// case sensitive off
TypeData<2, 7> = 10                         ;// 10 visible rows max
TypeData<2, 8> = 0                          ;// Don't fire the OnOptionClick
TypeData<2, 9> = 1                          ;// Reduce the list to partial matches
TypeData<2, 10> = 0                         ;// Only show the drop down when the user types
TypeData<2, 11> = 0                         ;// Do not use LIST Format
TypeData<2, 12> = 1                         ;// Autofill on first names
TypeData<2, 13> = 1                         ;// Hide dropdown when user clears cell
TypeData<2, 14> = 0                         ;// Let navigation keys show the drop down
TypeData<2, 15> = 0                         ;// Show the drop down regardless of the cell's contents
TypeData<2, 16> = 0                         ;// Show the drop down during autofill
TypeData<2, 17> = 1                         ;// Remove selection when user backspaces/deletes
TypeData<2, 18> = 0                         ;// Show Popup while in read only mode
```

```
TypeData<2, 19> = 1                         ;// Show Popup When Navigating
TypeData<2, 20> = 1                         ;// Always Tab Out on Enter
TypeData<2, 21> = -1                        ;// Always show the dropdown above when close to the screen bottom
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellType[1; All]", TypeData)
```

See Also

ImageList, CellCheck, CellCheckEnabled, CellColors, CellComboSelPos, CellComboRowData, CellComboData, HeaderType