

BeforeEdit

Fired just before a cell goes into edit mode.

Parameters

Parameter	Description
Cell	Cell about to go into edit mode
Reason	The reason why the cell is entering edit mode

Remarks

The BeforeEdit event fires when a cell is about to enter into edit mode. This event only fires for editable or read only cells since protected cells cannot enter edit mode. The Cell parameter references the cell entering edit mode, and the Reason parameter indicates the action that triggered edit mode in the first place.

Like all Before events, you can use prevent the cell from entering into edit mode by setting the [Cancel](#) property to 1. Unlike other Before events, however, there is no matching AfterEdit event. Instead, the [BeforeUpdate](#) and [AfterUpdate](#) events are fired when a cell is leaving edit mode.

Using the [Cancel](#) property to cancel the event only works if the event was qualified synchronously.

The Reason parameter can have many values depending on the action that led to edit mode being activated. Here are all the scenarios.

Double-Click Triggered Edit Mode

If a user double-clicks to enter into edit mode, then "Double Click" will be Reason value.

Function Key Triggered Edit Mode

Any of the 12 function keys can be assigned a task via the [FunctionKey](#) property. If a function key is used to trigger edit mode, then it will be passed via the Reason parameter. For example, using F4 to enter into edit mode will cause the Reason parameter to be set to "F4" as well.

Navigation Triggered Edit Mode

Navigation triggered edit mode occurs when the user navigates from one cell in edit mode to another cell, and the edit mode carries over. For example, if all cells' [CellEditMode](#) properties are set to "Edit", they go into edit mode as soon as they receive focus. Therefore, you might see one of the following values in the Reason parameter as a result.

Value	Description
Ctrl Down	The user used CTRL and the down arrow
Ctrl End	The user used CTRL and the END key
Ctrl Home	The user used CTRL and the HOME key
Ctrl Left Arrow	The user used CTRL and the left arrow
Ctrl Pagedown	The user used CTRL and the PAGEDOWN key
Ctrl Pageup	The user used CTRL and the PAGEUP key
Ctrl Right	The user used CTRL and the right arrow
Ctrl Up	The user used CTRL and the up arrow
Delete	The user deleted a record and caused the position to move to another existing record
Down	The user used the down arrow
End	The user used the END key
Enter	The user used the ENTER key
Home	The user used the HOME key
Left Mouse	The user used the left mouse button
Left Arrow	The user used the left arrow

New Record	The user created a new record and the position moved to the beginning of it
Pagedown	The user used the PAGEDOWN key
Pageup	The user used the PAGEUP key
Reset	The user left the table and the position reset to the top left cell
Right	The user used the right arrow
Shift Tab	The user used SHIFT and the TAB key
Tab	The user used the TAB key
Undo	The user changed position by undoing a previous cell edit
Up	The user used the up arrow

As you may have noticed, these are the same values that can be passed in the Cause parameter of the [PosChanged](#) and [PosChanging](#) events.

Typed Character Triggered Event Mode

Another way of entering edit mode is when a user starts typing in a cell that isn't in edit mode. In this case, Reason will be a single character, and that character represents the typed character that triggered edit mode. For instance, if the user types "a" while in navigation mode, then the cell will enter into edit mode and the Reason parameter will be set to "a" also.

Miscellaneous

There are a few other reasons that a cell might go into edit mode. Here are the other values you might get:

Value	Description
ComboDropDown	The cell is entering edit mode because it's combo drop down is appearing, whether by user action or by setting the ComboDropDown property
EditCell	Method The cell is entering edit mode because the EditCell method was called

Limitations

There is one limitation with this feature. The BeforeEdit event will not fire when the control gets focus. During tests, firing this event while getting focus would cause OI to remove and reset focus over and over until the control hung. Therefore, the logic for firing the event during GotFocus was removed. Instead, you can use the table's GotFocus event to capture edit mode, assuming of course your cells' CellEditMode properties are all the "Edit" mode.

Example

```
Transfer Param1 to Cell
Transfer Param2 to Reason

// When a user double clicks to edit a cell, display a custom edit window
If Reason _EQC "Double Click" then

    // Cancel the normal edit mode
    Call Set_Property(CtrlEntId, "OLE.Cancel", 1)

    // Use our own
    Call Show_Custom_Edit_Window(CtrlEntId, Cell)

end
```

See Also

[BeforeUpdate](#), [AfterUpdate](#), [PosChanged](#), [PosChanging](#)