

10Tec iGrid ActiveX 7.x

What's New in the Latest Builds

Keywords used to classify changes:

- [New] – a totally new feature;
- [Change] – a change in a member functionality or interactive behavior;
- [Fixed] – a fixed bug or solved problem;
- [Removed] – a member was completely removed;
- [Enhancement] – some functionality was enhanced;
- [Optimization] – a feature has speed improvements;
- [Renaming] – a member was renamed;
- [Code-Upgrade] – a special tag used to mark changes that may require changes in the source code for older versions.

v7.00, build 0015 | 2019-Mar-15

1. [Enhancement] Now iGrid does not remove selection from a cell if the row or column it belongs to becomes invisible. This feature allows you to implement filters and save the selection status for cells or whole rows in row mode when a row becomes visible/invisible during filtering. To provide the behavior expected by the user, the DELETE key and the CTRL+X/CTRL+C keyboard combinations do not process selected invisible cells.
2. [Enhancement] This version of iGrid guarantees that the **ClickNoDbClick** event is raised after the **MouseUp** and **Click** events.
3. [Fixed] Several problems related to the copy-paste functionality and the DELETE key were fixed (when iGrid had invisible rows/columns, when the row text column was visible, etc.).
4. [Fixed] If row mode was on and the visibility of a column changed, iGrid may have updated the selection status of the cells from this column in selected rows improperly.
5. [Fixed] If row mode and multi-select mode were on, iGrid did not remove selection from a group row when you cleared the **CellSelected** property for the row text cell of this row:

```
iGrid1.CellSelected(1, iGrid1.RowTextCol) = False
```

6. [Fixed] When iGrid had the input focus, the Escape key was processed entirely by iGrid and was not passed to the form's key handling events. As a result, the logic for the Escape key implemented in the form's **KeyDown** event or the form's button set as the cancel button did not work. In this build of iGrid the Escape key reaches the form's key handling events in all situations unless it does something in iGrid like canceling user input in a text cell or clearing the search string if incremental search is on.
7. [Fixed] The **Group** method failed if you calculated the SUM() or AVG() aggregate functions for non-numeric cell values.
8. [Fixed] Pausing on a break point inside an event handler of the **RequestCellToolTip** event crashed the development environment.
9. [Fixed] The **LayoutCol** property did not recalculate column widths properly for non-standard dpi settings.
10. [Fixed] iGrid may have drawn its contents incorrectly when the **DisplayMode** property of its scroll bars was set to **igScrollBarDisplayNever** or **igScrollBarDisplayAlways**.

11. [Fixed] The **Click** event was raised even if the mouse pointer was outside of the cell in which the mouse button was pressed, and the coordinates of the cell under the mouse pointer in which the button was released were passed in the event parameters. Now the **Click** event is raised only if the mouse pointer remains within the bounds of the cell in which the mouse button was pressed.
12. [Fixed] iGrid may have worked improperly if the input focus was moved to another window after pressing the mouse button in iGrid but before releasing it. For example, if multi-select mode was on and the user pressed the mouse button holding down the SHIFT key to select a range of cells, and then a message box was displayed from an event handler of the **CurRowChange** event, iGrid continued to select a range of cells after closing the message box as if the mouse button was still pressed. Another example of such a situation is when Windows brings another application to foreground.