

10Tec iGrid for .NET 6.x

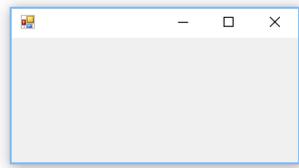
What's New in the Control

Tags used to classify changes:

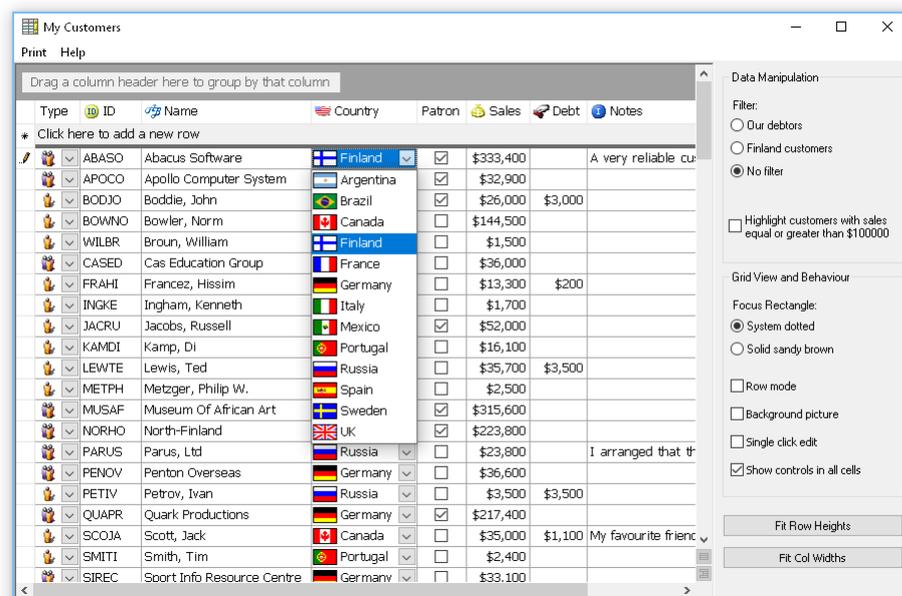
- [New] – a totally new feature;
- [Change] – a change in a member functionality or interactive behavior;
- [Fixed] – a fixed bug or solved problem;
- [Removed] – a member was completely removed;
- [Enhancement] – some functionality was enhanced;
- [Optimization] – a feature has speed improvements;
-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

v6.00, build 0018 | 2018-Aug-28

1. [Fixed][Enhancement] A serious problem related to the iGrid drop-down controls infrastructure was fixed. If the user opened a built-in drop-down list (**iGDropDownList**) or a custom drop-down control based on the **iGDropDownControl** interface, any new form created in the same app may have had a parasitic double shadow. This effect was especially noticeable in Windows 10:



The problem was related to the internal implementation of the WinForms package in .NET and the system shadow effect used for iGrid drop-down controls since Windows XP. The new implementation of shadows behind drop-down controls introduced in this update of iGrid is based on the modern Aero theme used in the latest versions of Microsoft Windows. An unobtrusive shadow similar to the shadow around top-level windows is displayed around drop-down controls in iGrid now, which creates a consistent look and feel with other windows in the OS:



The new implementation also solves one more problem with shadows under iGrid drop-down lists in its previous builds. If a dialog was displayed from an event handler of the **BeforeCommitEdit** event when the user selected an item in the list, the shadow behind the drop-down list disappeared while

the dialog was active. In the new build of iGrid this shadow remains but becomes less strong indicating the fact that the input focus was moved to another window.

2. [Fixed] The **iGRow.AutoHeight()** and **iGRowCollection.AutoHeight()** methods did not work correctly if the processed rows contained cells merged horizontally. The same problem was fixed for the **iGCol.AutoWidth()** and **iGColCollection.AutoWidth()** methods if columns contained cells merged vertically.
3. [Fixed] Changes in the **Header.SeparatingLine** property of one iGrid were copied to all other grids in the application. The same problem was fixed for the **Footer.SeparatingLine** property.
4. [Fixed] iGrid did not draw its rows correctly after moving or hiding/showing them.

v6.00, build 0012 | 2018-May-18

1. [Enhancement] The level indent area of a cell in the tree column is filled with the background color specified not only in the row cell style, but also in the cell's style or the column's cell style. The **ColorizeRowLevelIndent** property of iGrid now can be used to specify whether iGrid fills the level indent area in the cells of a tree column with the colors listed above.
2. [Fixed] If keys were specified for rows, iGrid threw an exception when the developer tried to move rows from code using the **iGRow.Move()** method or by changing the **iGRow.Index** property.
3. [Fixed] iGrid crashed when the user clicked the empty header area after the last column header.
4. [Fixed] iGrid did not draw the horizontal grid lines in the cells of the tree column properly after merging some cells in the grid.
5. [Fixed] Merged cells intersecting with selected rows in row mode were drawn using the individual cell selection colors defined in the **SelCellsForeColor** and **SelCellsBackColor** properties instead of the colors defined in the **SelRowsForeColor** and **SelRowsBackColor** properties.
6. [Fixed] If the **ShowControlsInAllCells** property was set to False, iGrid did not display the drop-down list when the cell combo button was pressed.
7. [Fixed] If you set background colors for odd and even rows with the **BackColorOddRows** and **BackColorEvenRows** properties of iGrid, merged cells spanning rows may have switched their background color during vertical scrolling of iGrid.
8. [Fixed] The **MergeCellsInCols** method failed for cells with null values.
9. [Fixed] If iGrid had invisible columns that had been moved, setting the **iGCell.Selected** property to True did not result in selecting the cell in some cases.
10. [Fixed] In apps compiled for .NET Framework 4.5 or higher, the filter box dialog of AutoFilterManager decreased in height and width by 8 pixels with every opening. The issue was related to the implementation of **SystemInformation.FrameBorderSize** in .NET that returns incorrect values in .NET 4.5 and higher versions.

To find out more about this issue in .NET 4.5+, read the following StackOverflow discussion:

<https://stackoverflow.com/questions/12856708/the-systeminformation-frameborder-size-return-wrong-size>

11. [Fixed] Reordering columns or rows with non-empty keys in the corresponding collection editors at design time led to the "Key already exists" error and corrupted column/row set at an attempt to save the changes.
12. [Fixed] The **TreeCol** and **SearchAsType.SearchCol** properties of iGrid could be set to incorrect values after changing the column set in the column collection editor at design time.