

NetAdvantage[®] for Windows Forms 12.1 Service Release Notes – April 2013



Add complete usability and extreme functionality to your next desktop application with the depth and breadth our Windows Forms UI controls. NetAdvantage for Windows Forms continues to make strides in user experience with the latest Windows[®] 7 and Office[®] 2010 styles.

Installation

Downloading

[Download NetAdvantage for Windows Forms here.](#)

What's Changed

Components	Product Impact	Description
Infragistics excel Engine	Bug Fix	Excel library displays incorrect value when using formula. Note: Fixed an issue in the Excel library that was displaying the value incorrectly for numeric cells when the culture had a character other than a dot (.) as the decimal separator.
Infragistics excel Engine	Bug Fix	Row Row height gets slightly bigger on the exported Excel if cell.CellFormat.Fill is changed. Note: Fixed an issue with the Excel library where the heights of certain rows were incorrect when loading a workbook from a file..
AppStylist	Bug Fix	ReadOnly style defined in AppStylist is applied to a cell when the cell is activated.
WinChart	Bug Fix	System.ArgumentOutOfRangeException is thrown for histogram chart depending on TickmarkInterval and RangeMin/Max settings.
WinNumericEditor	Bug Fix	Unable to define a string when the mask property is specified. Note: In order to define a string while the mask property is specified, you need to set the DataMode on the UltraMaskedEdit control to Raw. This allows the control to skip the literals when you assign the Text or Value properties. Instead of changing the default behavior we added a new static property to correct the behavior. So in addition to setting the DataMode to Raw, you must also set: Infragistics.Win.EditorWithMask.MaskDataModeRawSetValueSkipsLiterals = True This property can be set once and will affect all MaskedEdit controls in the application.

WinGrid	Bug Fix	<p>Filter dialog does not remember previous selections.</p> <p>Note: The UltraGridFilterUIProvider will now attempt to match up derived FilterCondition classes when they are assigned as the DataValue of an item in the FilterDropDownList. However, since FilterConditions are cloned, this cannot be done via a reference comparison. Instead, the derived FilterCondition class must override GetHashCode and Equals in order to indicate that it is equal to another FilterCondition with the same criteria.</p> <p>In a simple case, the implementation might look like this:</p> <pre> public override bool Equals(object obj) { CustomFilterCondition objCustomFilterCondition = obj as CustomFilterCondition; if (null == objCustomFilterCondition) return false; if (this.CompareValue == objCustomFilterCondition.CompareValue && this.ComparisonOperator == objCustomFilterCondition.ComparisonOperator) { return true; } return base.Equals(obj); } public override int GetHashCode() { return this.CompareValue.GetHashCode() this.ComparisonOperator.GetHashCode() << 8; } </pre>
WinExpandableGroupBox	Bug Fix	First chance exception has been thrown when accessing the 'Panel'.
WinTree	Bug Fix	NullReferenceException is thrown when calling the Reposition() method on an UltraTreeNode.

WinTree	Bug Fix	ArgumentOutOfRangeException occurs when the active node is the last (bottom) node, and its Visible property was changed false.
WinTree	Bug Fix	Pressing the SpaceBar changes the CheckState for nodes with a space as its initial character.
WinMessageBoxManager	Bug Fix	Pressing Alt+F4 closes the MessageBox with incorrect DialogResult (cancel).