

About Column Resizing

DataView has many options available to customize column resizing behavior. It is important to understand the options and how they work and don't work together.

ColumnResizingMode Controls how the columns are resized when the DataView/window is resized. (None, Distributed, LeftToRight, RightToLeft)

Using a **ColumnResizingMode** other than None causes the columns to be reduced to prevent columns from expanding beyond the width of DataView and is thereby incompatible with **Push ColumnUserResizingMode**. When all minimumWidths are reached, horizontal scrolling is enabled **unless AllowsOverflow=false**, in which case columns are reduced proportionally to fit the view.

ColumnUserResizingMode controls how the columns are resized when the user drags between two columns. (Shared or Push) **Push** causes the resizing column to change width, while the columns to the right are moved to make space for the new width. **Share** causes the column to right of the resized column to be reduced/grown to compensate for the new width.

In Xojo, the listbox control allows a few different ways of handling column resizing (literal widths, % widths and * widths). These are valid in DataView, but not compatible with some options.

For example, **using * values for columnWidthExpression** will fill the remaining space after other columns have been sized. This is **not** compatible with **PUSH userColumnResizingMode**, and **Not** compatible with any of the **general columnResizingModes** (Distributed, LeftToRight, RightToLeft)

Using **% Values for columnWidthExpression** is compatible with **PUSH userColumnResizingMode** but **not** compatible with the **general ColumnResizingModes**.

Literal Values (point sizes) are compatible with all modes.