Mergable Cell Listbox
Overview/Demo Documentation
Version 1.0

Karen Atkocius
keatk@verizon.net

Overview

The listbox subclasses classes in this project add several useful new features to the standard REALStudio listbox. In my past life often had to create tables to display/report/manage data. My tool of choice was a spreadsheet as it was both easy to do calculations and lay out tables that were easy to read/understand. One of the most useful tools for creating easily readable layouts was the ability to merge cells to organize data logically, clearly. and use space efficiently.

That was what I missed most when I tried to use the REALStudio listbox for data presentation. This project is the result of that frustration. The main focus of these classes is to provide as comprehensive as possible support for merging cells across rows and columns to be able to do the type of data presentation one can do in Excel.

An area of merged cells for the most part is treated as a single cell having the coordinates of the top left cell of the group. See the API documentation for details. When a row is selected all multi-row merged cells which intersect it are highlighted. Be aware that supporting this highlighting in a consistent manor necessitates a lot of listbox redrawing.

The features specific to merged cells are:

1) Merging and unmerging of cells across both rows and columns
2) Column resizing across merged areas
3) Editing merged cells
4) Rows may be inserted or deleted within merged area that cross rows. The merged area is either expanded or contracted as expected
5) Rows which are folders may be merged as long as the folder is the top row or last row

In addition other useful features are also included:

1) Specifying a background color as well as alternating row colors via the property pane
2) Support for Multi-line text a cell:
   - Text Wrapping. Does **NOT** simply use g.DrawString with the WrapWidth parameter. Only displays lines that are fully visible and if all text is not displayed last line will have an ellipsis even if that line fits in the cell (ends a paragraph)
   - Editing Multi-Line text (can use return key to start new line)
   - May specifying line spacing and line by line horizontal alignment
   - May optionally use a vertical scrollbar for in cell editing
3) Allows aligning text (single or multi-line) vertically as well as horizontally
4) Provides enhanced cell text attribute and border assignment methods
5) Hierarchical Listbox support:
   - Uses the standard RealStudio listbox hierarchical API. Does not force a specific data structure
   - Optionally draws lines connecting folders and their children
   - Enhanced UI for Expanding/Collapsing Folders:
     • Option Click on a closed disclosure widget will open all sublevels
     • SpaceBar toggles disclosure widget. If closed Option-SpaceBar will open all sublevels
     • Option-UpArrow closes an open folder. Option-DownArrow opens a folder. If closed, Shift-Option-DownArrow will open all sublevels
   - Provides methods to:
     • determine if a row is a folder
     • return the indent level of a row
     • open all sublevels of a folder
     • close all open folders
     • inhibit the drawing of the horizontal connector line to a child row. Useful for visual grouping when related data is presented in multiple child rows at the same indent level. In other words data for one logical node is displayed in multiple consecutive rows. (See Lot Testing Demo)

All features work on OS X and Windows (only XP tested).

At this time on Linux the CellBorder settings do not work as expected . This may be worked around by drawing borders in the paint events. Also drawing connecting lines in hierarchical listboxes does not work for cells merged across rows. In addition on Linux, the CellPaint events can sometime draw outside of their cells and outside the listbox (a REALStudio bug), particularly when resizing columns or the listbox. Also on Linux resizing can also be very sluggish and unattractive visually.

Merged cells may be able to be used in Linux in some specific cases, however the potential for visual glitches is high. Ubuntu screen screenshots are included for all demo screens.

The demos included were created to test/debug the classes, as well as show possible uses and demonstrate how to code them. This assumes you have good understanding of the standard REALStudio listbox API.

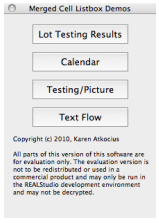
I tried to design the extensions to be as consistent as possible with REAL's listbox API design (which is very flexible but not simple) as I could. However because a merged cell may cross both rows and columns one must take that into consideration when coding using the standard listbox APIs.

For example if a cell is merged across rows, in an event, just because listbox.Selected(row) is false does not mean a cell the row intersects is not selected! You would need to check using the new method SelectedCell(row,column) to be sure.

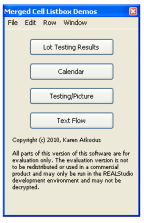
The Demo Application

When you startup the demo app you will see a window showing the demo options:

Mac OS X



Win XP

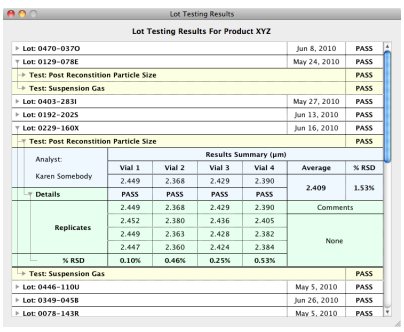


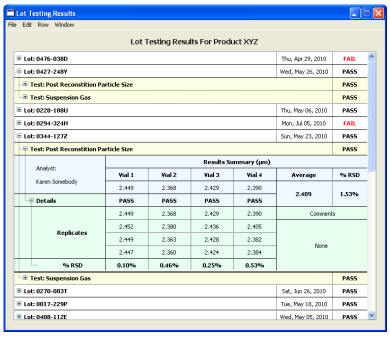
Ubuntu

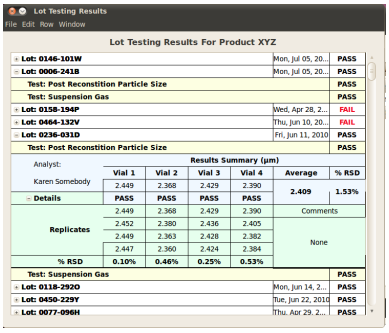
There are 4 Demo windows. The are accessible either through this dialog or through the Window menu.

1) **Lot Testing Results**

The "Lot testing Results" screen is meant to show how lot testing could be presented using merged cells and is shown below:







Each lot has 4 level each with more detail than the last. Merged cells are used to visually group the data within each level to maximize screen usage and improve readability for busy end user. Note that option clicking on a closed disclosure widget will open all of it's sublevels. This is not specific to the screen but supported generally by the base class.

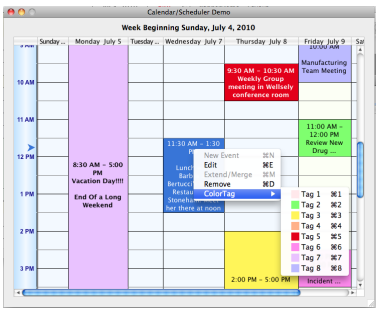
Normally when node lines are used there are lines connecting a folder to each of it's child rows. In the case of merged cells there is only one connect to the top merged row.

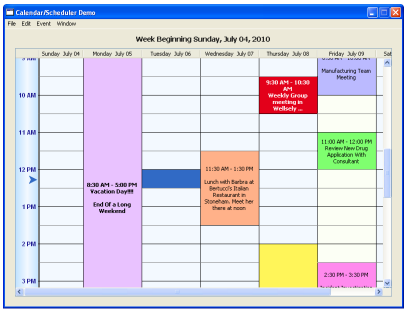
But notice that there is no connection from the "Test: Post Reconstitution …" to the child "Analyst: …." or from the "Details" folder to the child "Replicates". This is because of the use of the InhibitConnector method which prevented them from being drawn. This gives a better visual grouping in this case.

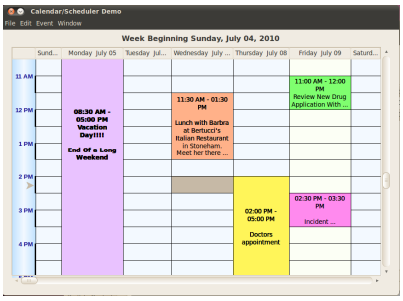
At the lowest level, the comments fields is editable. Clicking on it will allow you to enter text directly into it. The window (and so the listbox) is resizable.

2) **Calendar**

This demo of a daily schedule planner app and is more extensive. Examples shown below:







Here the listbox selection type is set to multi-line and visually then selection is by cell. The mouse or the arrow keys may be used for navigation. Multiple selections are made using the shift modifier key with the keyboard or mouse. Drag selection can also be used. Only contiguous multiple cell selections are only allowed within a single column (day). Using the Control Key to make discontiguous selections in not allowed.

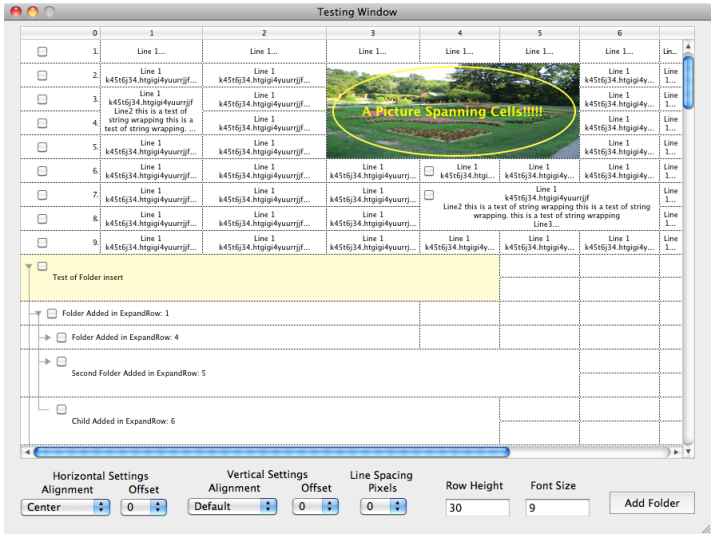
A selected cell or group if cells may used to create new events or extend existing events by either hitting return, selecting "Extend/Merge" or "New" from the Event menu , by right clicking and selecting the desired option from the contextual menu or using the appropriate menu shortcut. If a single event is selected hitting return or using the Event Menu (or contextual menu option) allows you to edit the text. Deleting an event may only be done via either of the menus mentioned or the menu shortcut.

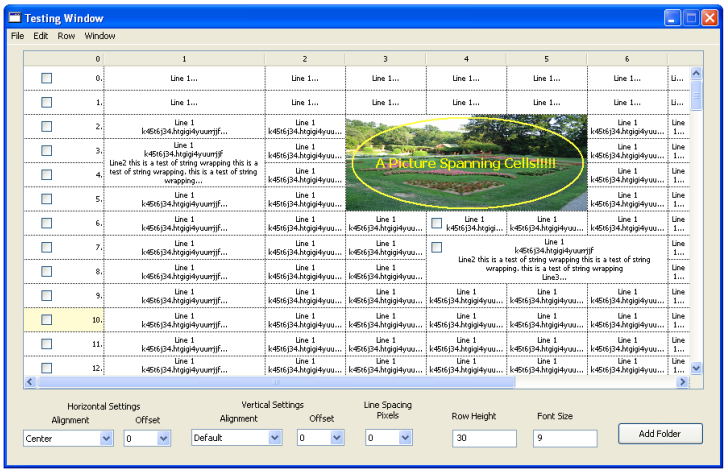
Selected cells may be "tagged" with a predefined color via the "Tag" submenu of the Event or contextual menu or the menu shortcut (MenuModifierKey + 1-8)

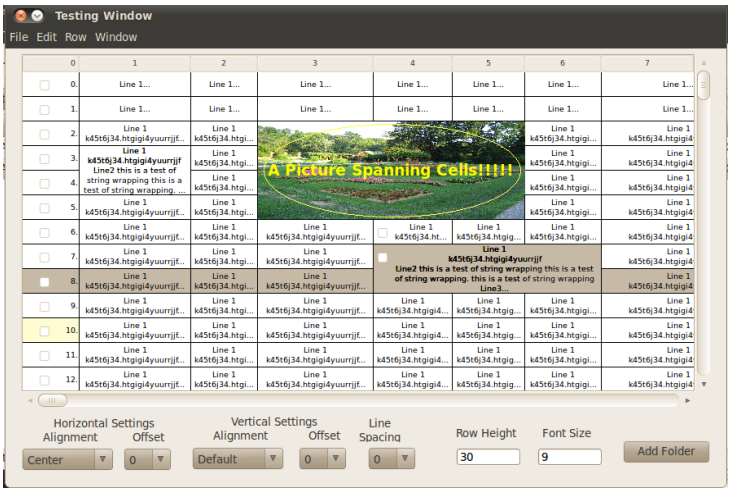
Columns are resizable, as is the window and with it the listbox.

3) **Testing/Picture Demo Window**

This is the window that was used to debug the listbox. It shows a picture as the background of a merged cell. The primary focus was to test/debug the visual aspects of cell merging, including selection, column resizing, editing, text wrapping, text alignment and checkboxes during development and merged folders. Examples from each OS are shown below:





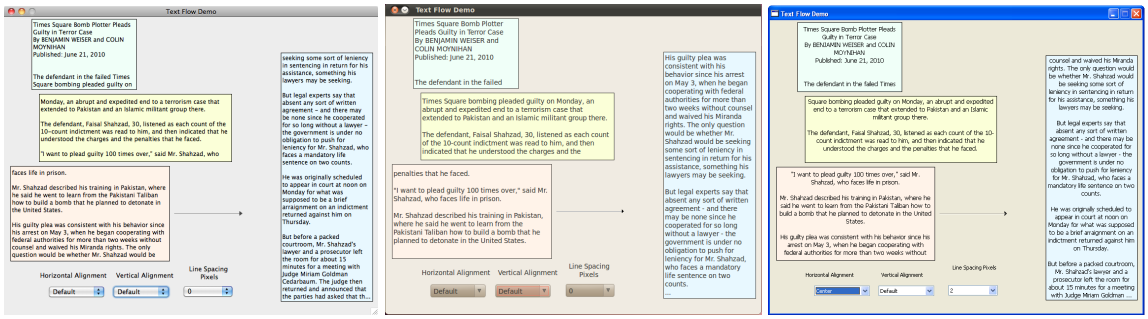


The popup menus allows changing test alignment and line spacing. The text fields allow changing row height and font size. Enter a value and hit return. The “Add Folder” button ad a folder, and merges it with another row. Clicking on the resulting widget shows child rows, some of which are also merged. This is shown in the OS X Screen shot.

If you edit the merged cell at row 7 column 4, the cell will have a vertical scrollbar while editing. The appearance of a scrollbar depends on the setting of the VerticalScrollBar property the TextArea subclass instance(s) placed on the listbox and the new event EditMode Event. See API documentation for details. The use of the details could have been hidden by the use of a Container control, but this would have limited use to Pro and Enterpise versions of REALStudio. Also this apprach

4) **Text Flow**

This is to demonstrate the text wrapping utility routines. While not part of the listbox classes they may be used in the paint events. However they can also be used to ‘flow’ a single large string into multiple rectangular areas for layout type text drawing. This is an example of that use:



The popup menus may be used to change the vertical alignment within each block, the line by line horizontal alignment of the text and the line spacing. The window may be resized somewhat changing the size of blocks, and the text will flow into the new dimensions.