20th September 2009 Glade3 tutorial (4) - GtkTreeView data backend

Welcome.

First, I need to tell you that we won't be connecting any callbacks today. I knew I promised this last time, but we simply don't have any data to manipulate inside callbacks. So what are we going to do today? Well, adding data seems like a good idea;)

Contents

Model-view-controller (MVC) design

Before we start doing anything, we need to know some things about how GtkTreeView operates. Information about this can be found in API documentation [http://scentric.net/tutorial/treeview-tutorial.html] and in GtkTreeView tutorial [http://scentric.net/tutorial/treeview-tutorial.html] . I would advice you to read those two references if you're just starting to code with GTK+.

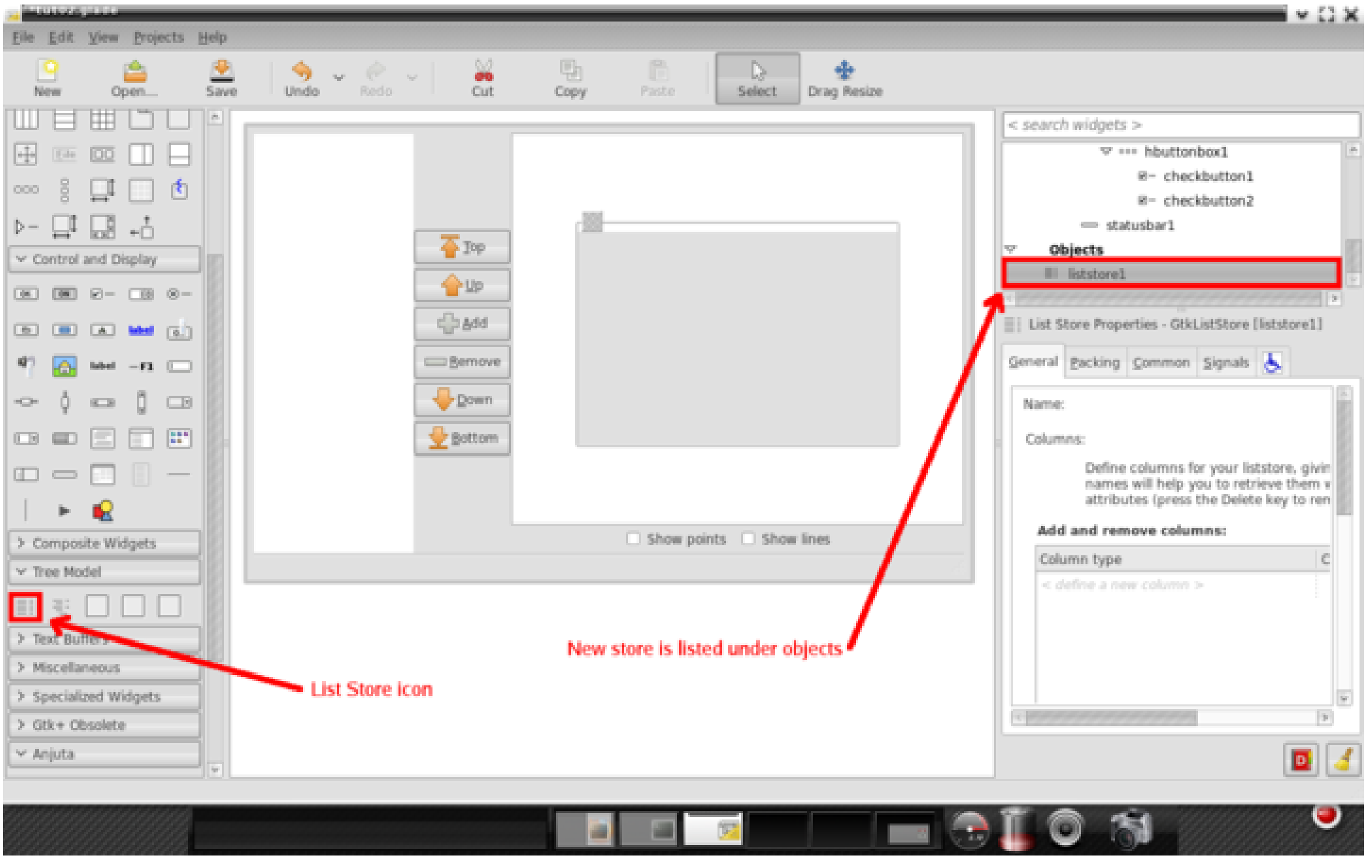
If you don't have time to read those two in entirety, I'll just quickly sum the contents:

1. GtkTreeView is basically just container that holds columns and provides surface on which cell renderers draw on.
2. GtkCellRenderer is object that draws inside tree view based on the data inside backend storage.
3. GtkTreeModel is interface that any data store needs to implement if it wants to be used as a data backend to tree view.
4. GtkTreeStore and GtkListStore are two data stores that implement GtkTreeModel interface and are part of GTK+ itself.

Keep this in mind since it's crucial for understanding what exactly will we be doing with glade.

Creating data backend

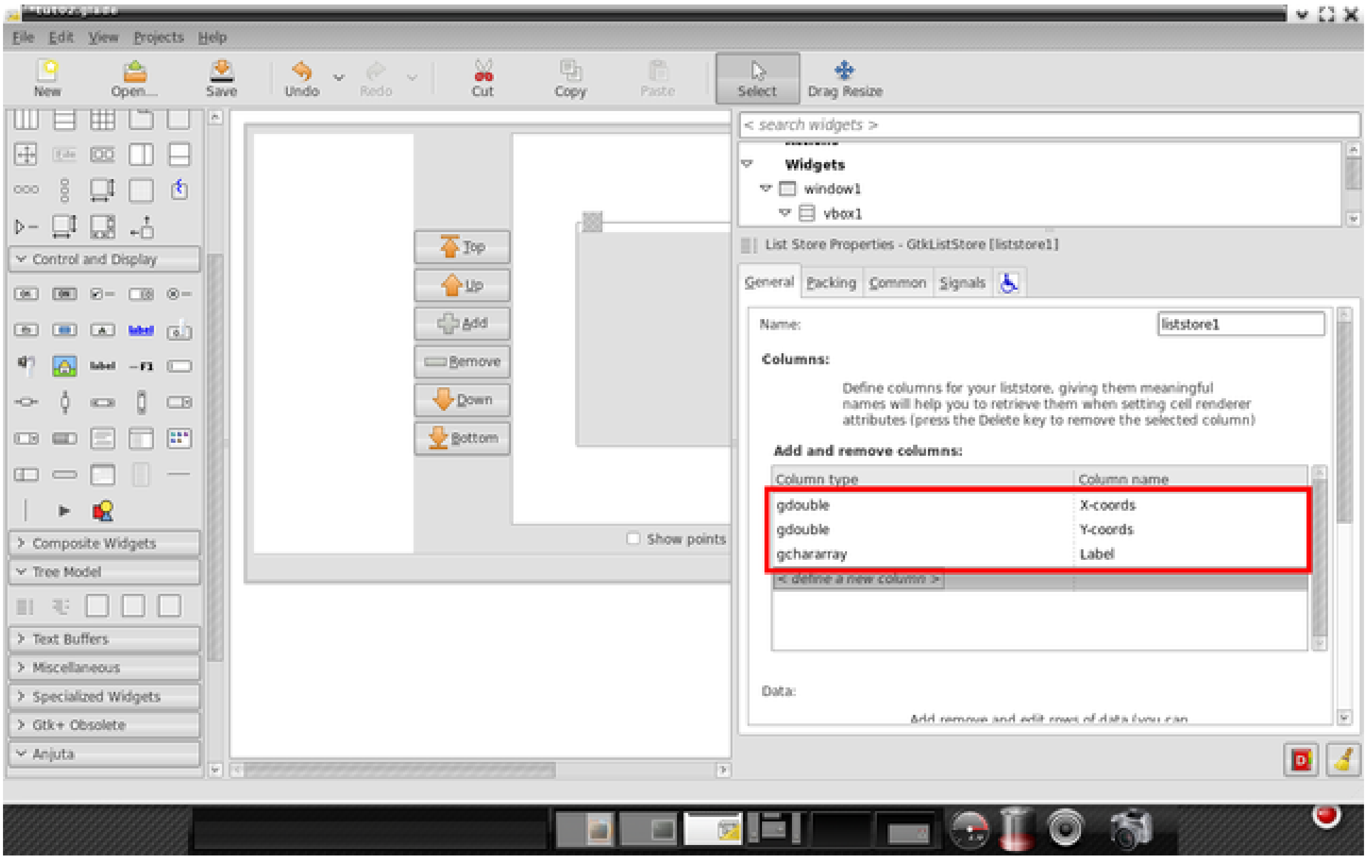
Load project form last post into Glade. Now scroll down widget catalog until you see "Tree Model" part. Expand it and click on "List Store" icon. New entry will appear under "Objects" inside object tree.


File Edit View Projects Help
New Open... Save Undo Redo Cut Copy Paste Select Drag Resize
Control and Display
Composite Widgets
Tree Model
Text Buffers
Miscellaneous
Specialized Widgets
Gtk+ Obsolete
Anjuta
Top
Up
Add
Remove
Down
Bottom
Show points
Show lines
< search widgets >
hbuttonbox1
checkbutton1
checkbutton2
statusbar1
Objects
liststore1
List Store Properties - GtkListStore [liststore1]
General Packing Common Signals
Name:
Columns:
Define columns for your liststore, givin names will help you to retrieve them w attributes (press the Delete key to ren
Add and remove columns:
Column type
< define a new column >
New store is listed under objects
List Store icon

You'll probably want to resize right panel to get more space for editing list store properties. We'll do two things now:

1. Define number of columns inside list store and their types
2. Add some sample data to list store for demonstration purposes

Number of columns and their types are defined in upper part of the "General" tab. Simply start typing into proper field to add new column. What columns do we need? Since we're developing charting application, we'll need one column that will hold X coordinates of the points, one column to hold Y coordinates and one column for optional label that can be added to point. Types of the columns will be gdouble for X and Y coordinates and gchararray for label column (gchararray is just another name for "gchar *" that is registered using GType system).



Name: liststore1

Columns:

Define columns for your liststore, giving them meaningful names will help you to retrieve them when setting cell renderer attributes (press the Delete key to remove the selected column)

Add and remove columns:

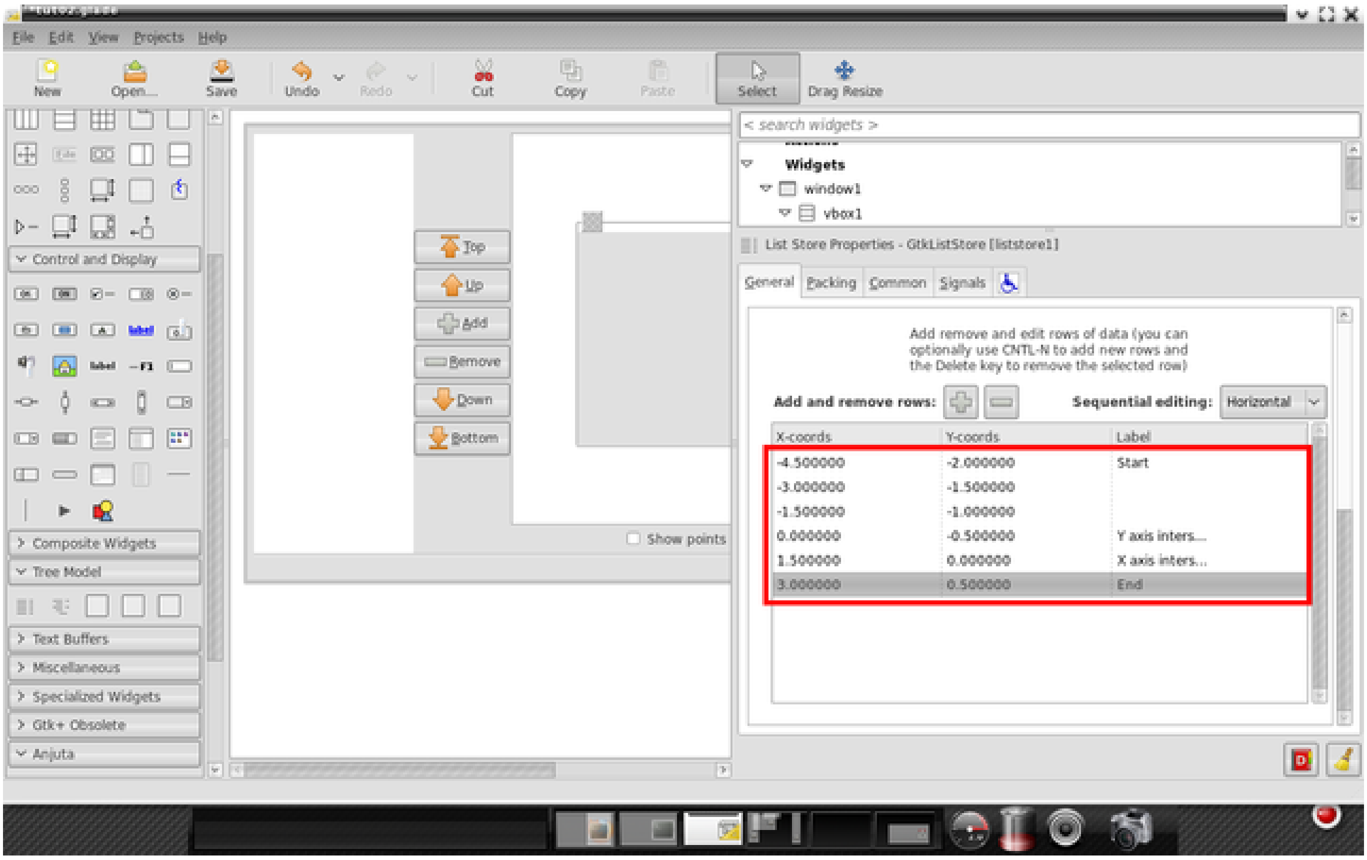
Column type	Column name
gdouble	X-coords
gdouble	Y-coords
gchararray	Label
< define a new column >	

Data:

Add remove and edit rows of data (you can

Now we'll add some sample data to list store. Scroll down to expose data insertion part, add six rows using plus button and fill them with with data like this:

```
+--------+--------+----------------------+
| - 4.5  | - 2    | Start                |
| - 3    | - 1.5  |                      |
| - 1.5  | - 1    |                      |
|   0    | - 0.5  | Y axis intersection  |
|   1.5  |   0    | X axis intersection  |
|   3    |   0.5  | End                  |
+--------+--------+----------------------+
```



List Store Properties - GtkListStore [liststore1]

Add remove and edit rows of data (you can optionally use CNTL-N to add new rows and the Delete key to remove the selected row)

Add and remove rows:

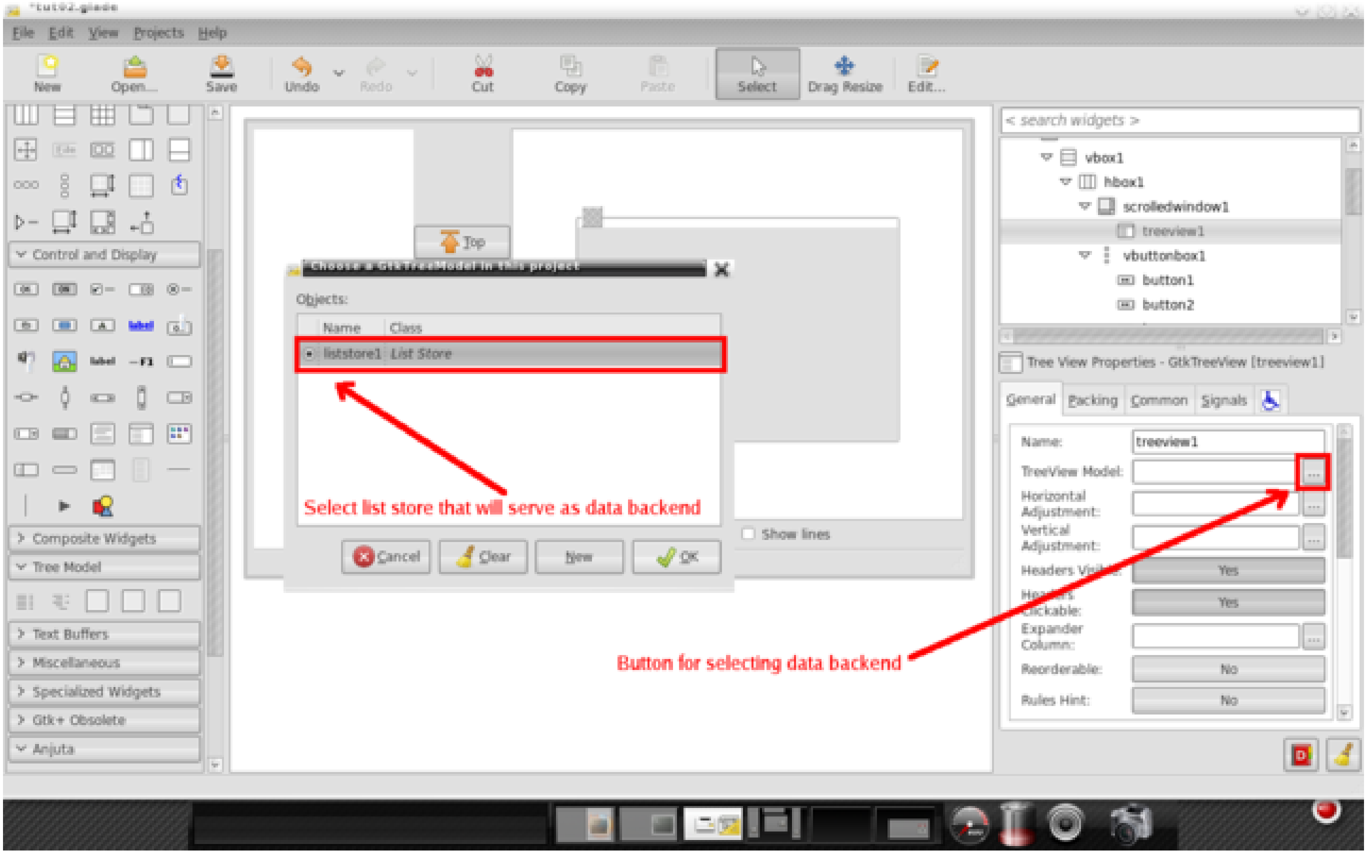
Sequential editing: Horizontal

X-coords	Y-coords	Label
-4.500000	-2.000000	Start
-3.000000	-1.500000	
-1.500000	-1.000000	
0.000000	-0.500000	Y axis inters...
1.500000	0.000000	X axis inters...
3.000000	0.500000	End

Our list store is finished now.

Connecting tree view with list store

Next, we need to connect tree view and list store. Select tree view and in "General" tab click button with "..." next to the "TreeView model" field. Inside pop-up dialog select "liststore1" and click "OK".



Choose a GtkTreeModel in this project

Objects:

Name	Class
liststore1	List Store

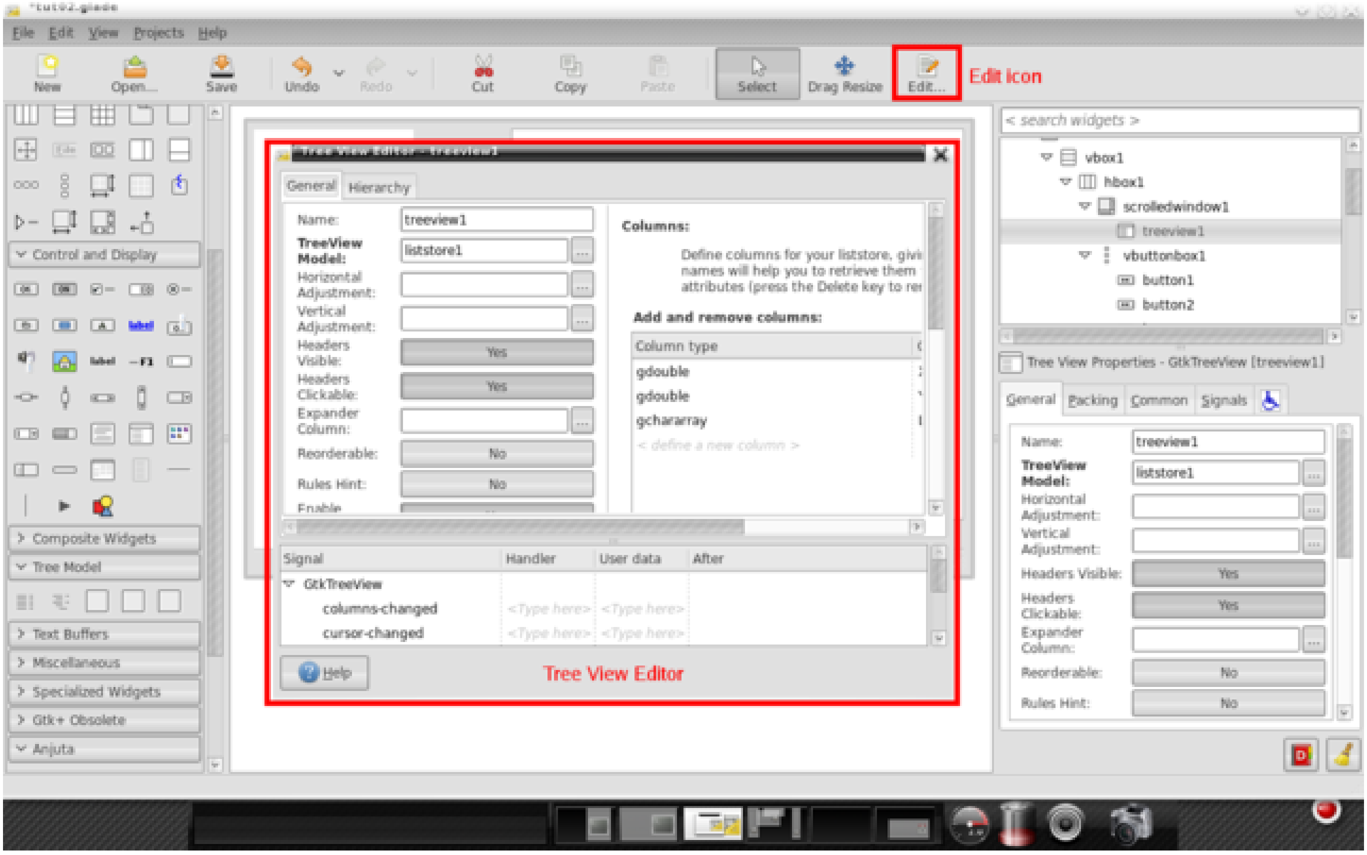
Select list store that will serve as data backend

Name: treeview1
TreeView Model:
Horizontal Adjustment:
Vertical Adjustment:
Headers Visible: Yes
Headers Clickable: Yes
Expander Column:
Reorderable: No
Rules Hint: No

Button for selecting data backend

Adding display components to tree view

Now we need to add columns to out tree view that will hold cell renderers. When we select tree view, new icon labeled "Edit..."will appear at the end of the toolbar. Clicking it will open new window with tree view editor.



Edit icon

Name: treeview1
TreeView Model: liststore1
Horizontal Adjustment:
Vertical Adjustment:
Headers Visible: Yes
Headers Clickable: Yes
Expander Column:
Reorderable: No
Rules Hint: No
Enable

Columns:

Define columns for your liststore, givi
names will help you to retrieve them
attributes (press the Delete key to re

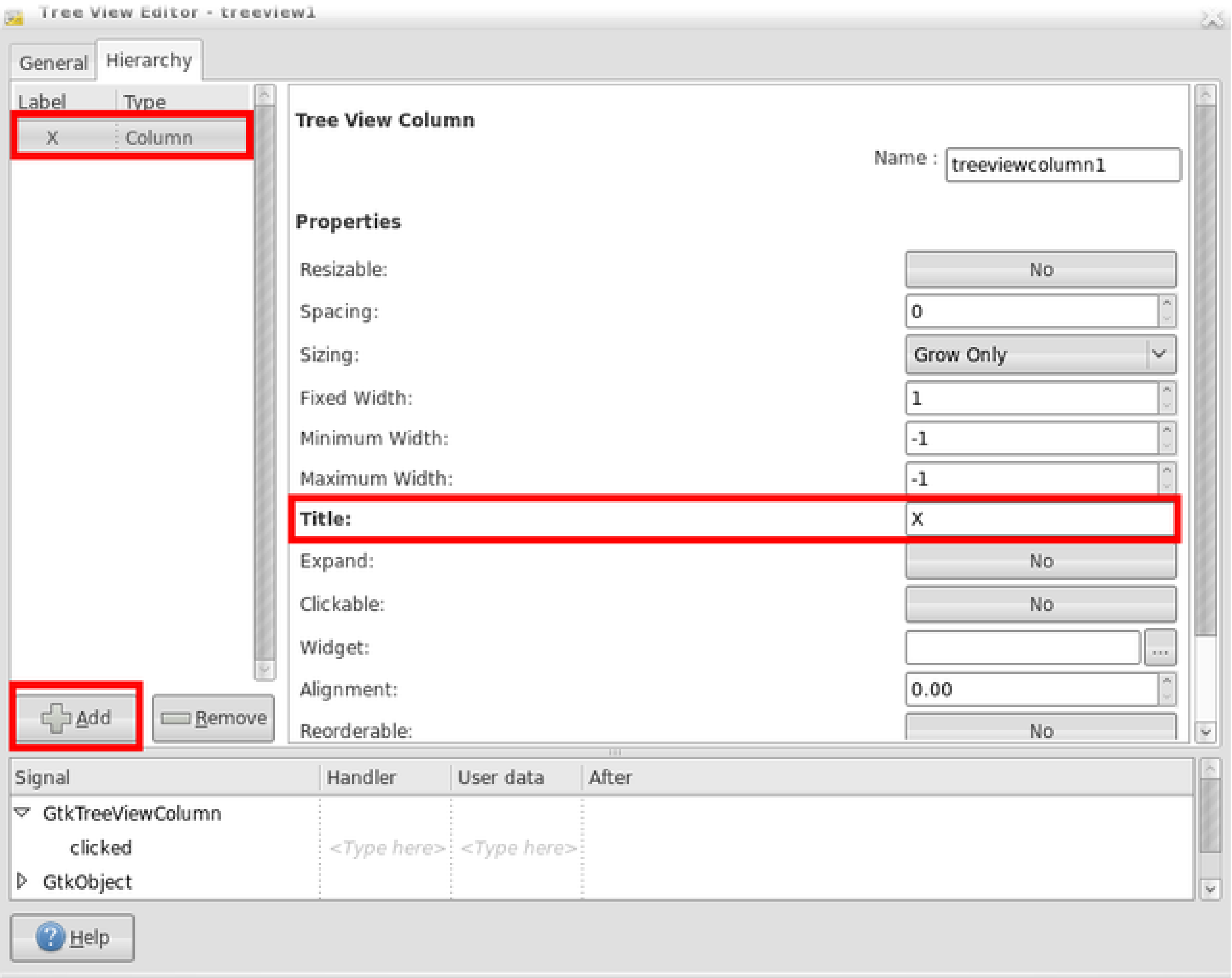
Add and remove columns:

Column type
gdouble
gdouble
gchararray
< define a new column >

Signal	Handler	User data	After
GtkTreeView			
columns-changed	<Type here>	<Type here>	
cursor-changed	<Type here>	<Type here>	

Tree View Editor

Inside "General" tab of Tree View Editor we can see some information we already entered before. To create display parts of tree view, we need to switch to "Hierarchy" tab, where we're presented with a lot of empty space;). To add new column, click "Add" button and new column will be added to left field, with it's properties displayed in rigth field. We'll change column's title to "X" and this is all that we'll do with this column.



Label	Type
X	Column

Tree View Column

Name : treeviewcolumn1

Properties

Resizable:	No
Spacing:	0
Sizing:	Grow Only
Fixed Width:	1
Minimum Width:	-1
Maximum Width:	-1
Title:	X
Expand:	No
Clickable:	No
Widget:	
Alignment:	0.00
Reorderable:	No

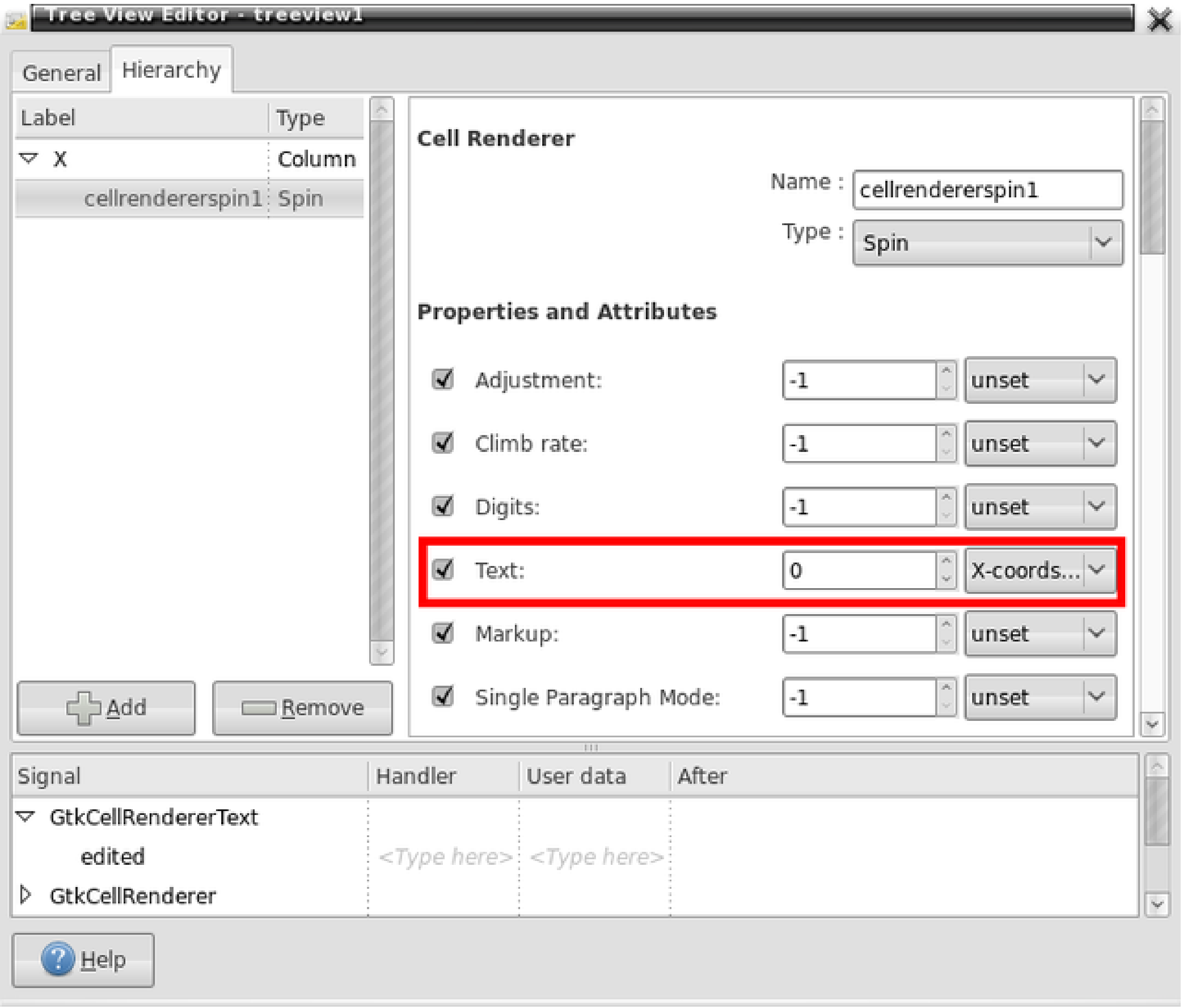
Signal	Handler	User data	After
GtkTreeViewColumn			
clicked	<Type here>	<Type here>	
GtkObject			

Now right-click newly created column and select "Add child Spin item", which will add new GtkCellRendererSpin to hierarchy. Now we need to again change some properties. But since changing cell renderer's properties is a bit different that changing properties of normal objects, I'll explain this a bit more in detail.

If you look at the properties editor, you'll notice that it is composed from 4 columns: first column holds check button, second column holds property name, third column holds spin button and last column holds combo box. Why are things so complex? Because cell renderers can have their properties set on two different ways: on a global basis, which means that this property will be the same for all lines that cell renderer draws; or on a per-line basis, which means that property is stored inside data backend and is read from there for each line.

How do these two methods map onto property editor? Check button controls whether property is set per-line or global (active check button means per-line, inactive means global). When we set some property on a per-line basis, we must inform cell renderer in which column of data store are values for this property stored, and we can use spin button to directly input column number or combo box to select column based on label that we assigned to it when constructing list store. When we set some property on a global scale, spin button and combo box are hidden and we get regular property editing widget instead.

Now let's set up our spin cell renderer. First property that we're goint to set is "Text". We want to display values from data backend, so we'll leave the check button activated and set the column, from which this property will take it's values to X-coords column.



Label	Type
X	Column
cellrendererspin1	Spin

Cell Renderer

Name : cellrendererspin1

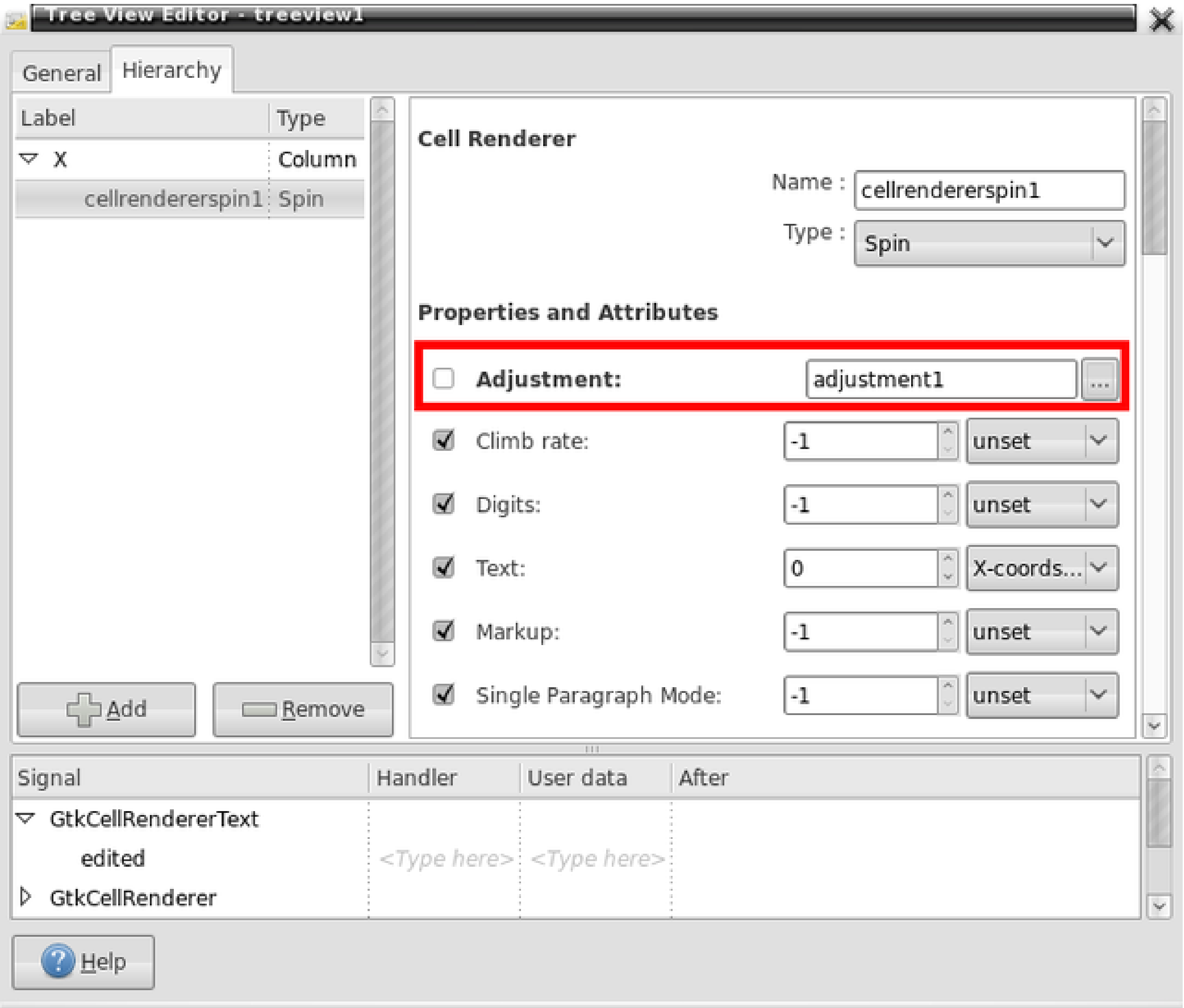
Type : Spin

Properties and Attributes

☑	Adjustment:	-1	unset
☑	Climb rate:	-1	unset
☑	Digits:	-1	unset
☑	Text:	0	X-coords...
☑	Markup:	-1	unset
☑	Single Paragraph Mode:	-1	unset

Signal	Handler	User data	After
GtkCellRendererText			
edited	<Type here>	<Type here>	
GtkCellRenderer			

Second property that we'll set is "Adjustment" property. We'll set it on a global scale, since we want all lines to have the same range of available values. So we need to remove tick from check button and then click the button with "..." and create new adjustment (we don't have any yet, so we created it).



Label	Type
X	Column
cellrendererspin1	Spin

Cell Renderer

Name : cellrendererspin1

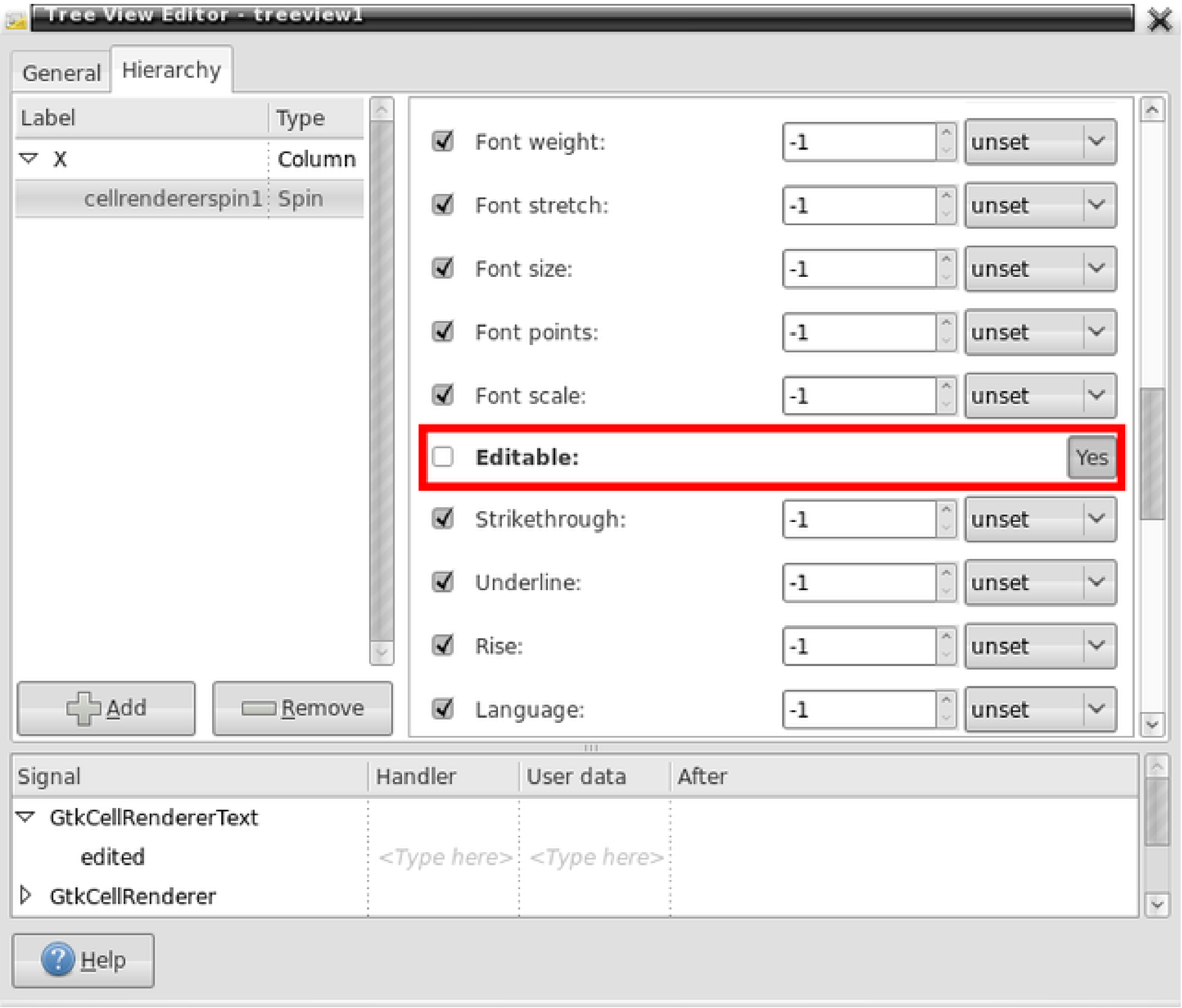
Type : Spin

Properties and Attributes

□	**Adjustment:**	adjustment1	
☑	Climb rate:	-1	unset
☑	Digits:	-1	unset
☑	Text:	0	X-coords...
☑	Markup:	-1	unset
☑	Single Paragraph Mode:	-1	unset

Signal	Handler	User data	After
GtkCellRendererText			
edited	<Type here>	<Type here>	
GtkCellRenderer			

Last property that we'll set is "Editable". We'll again set it globaly to "Yes", since we want all our X coordinate cells to be editable.



Label	Type
X	Column
cellrendererspin1	Spin

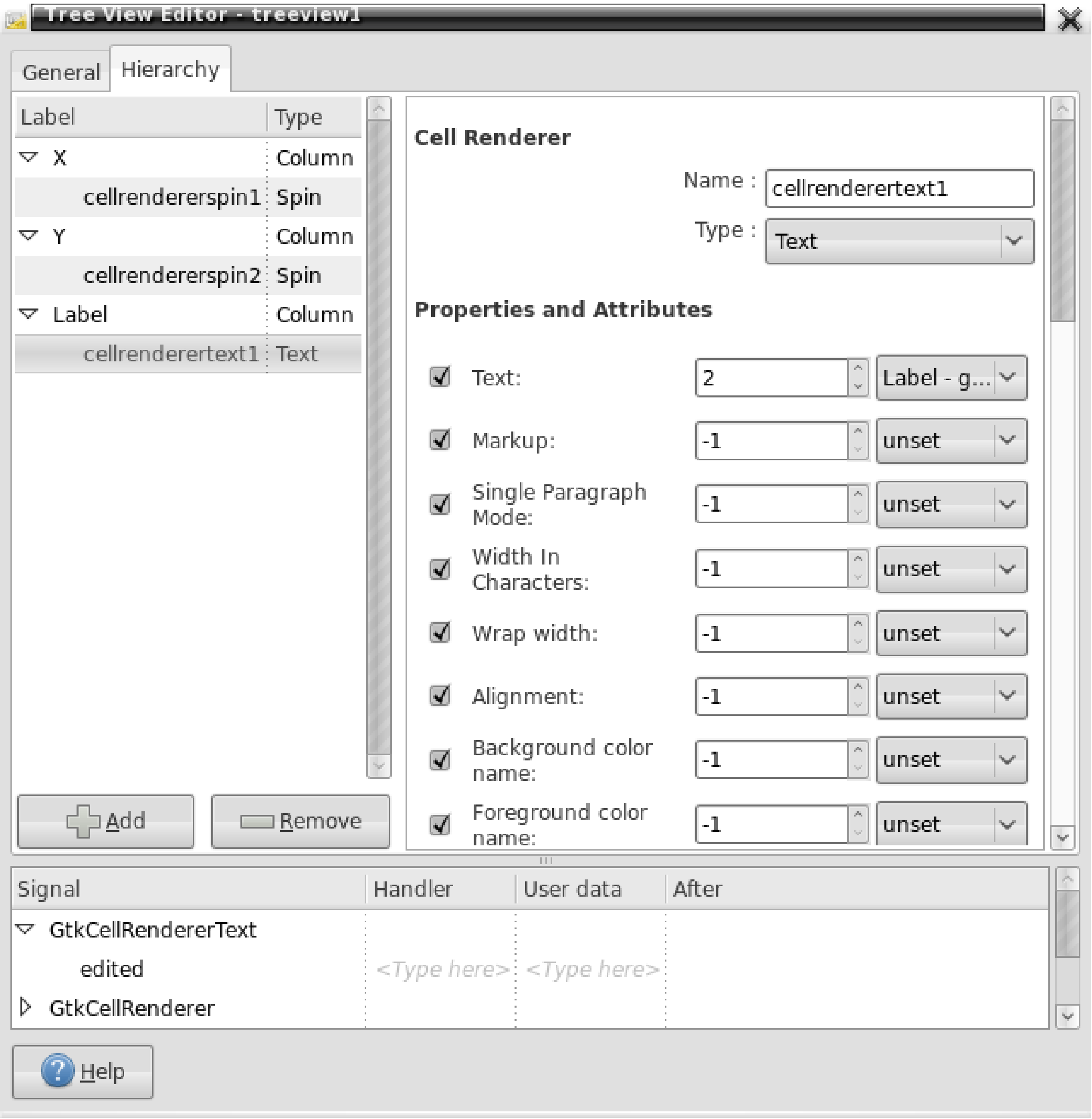
☑	Font weight:	-1	unset
☑	Font stretch:	-1	unset
☑	Font size:	-1	unset
☑	Font points:	-1	unset
☑	Font scale:	-1	unset
□	**Editable:**		Yes
☑	Strikethrough:	-1	unset
☑	Underline:	-1	unset
☑	Rise:	-1	unset
☑	Language:	-1	unset

Signal	Handler	User data	After
GtkCellRendererText			
edited	<Type here>	<Type here>	
GtkCellRenderer			

Now we need to create another column for Y coordinates. Process is exactly the same as before:

- right-click X column and select "Add Column item"
- change title to "Y"
- right click Y column and select "Add child Spin item"
- Adjust properties exactly as before (when setting "Text" property, slect Y-coords clumns as source of values and when you set "Adjustment" property, you don't create new one, just reuse adjustment1 that we created for X spin renderer)

Last column that we need to add will hold point labels. We again right-click on column Y and select "Add Column item". We set it's title property to "Label". Now right-click on Label columns and select "Add child Text item&qout; On the text renderer, set "Text" property to Label column adn "Editable" property globaly to "Yes".



Label	Type
X	Column
cellrendererspin1	Spin
Y	Column
cellrendererspin2	Spin
Label	Column
cellrenderertext1	Text

Cell Renderer

Name : cellrenderertext1

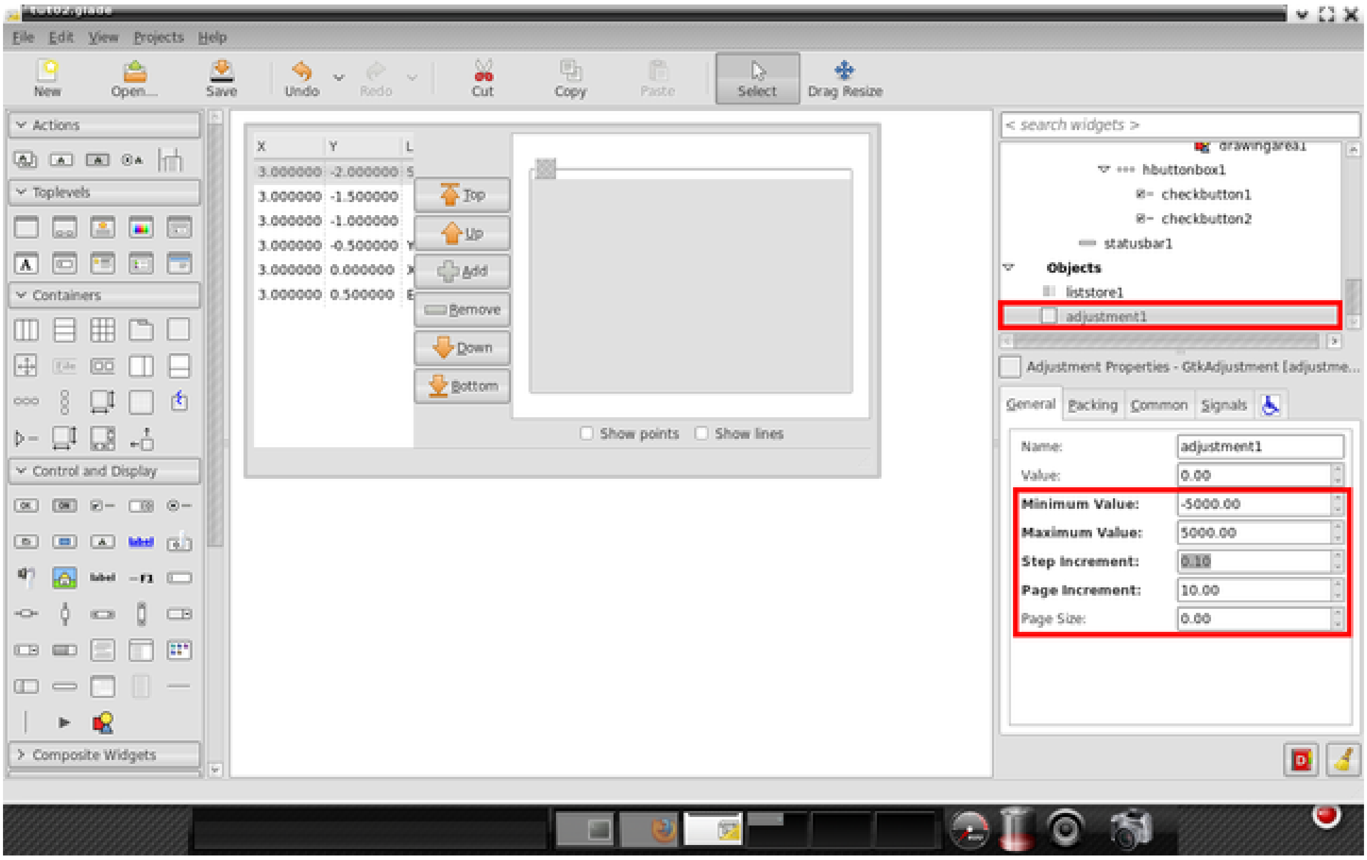
Type : Text

Properties and Attributes

☑	Text:	2	Label - g...
☑	Markup:	-1	unset
☑	Single Paragraph Mode:	-1	unset
☑	Width In Characters:	-1	unset
☑	Wrap width:	-1	unset
☑	Alignment:	-1	unset
☑	Background color name:	-1	unset
☑	Foreground color name:	-1	unset

Signal	Handler	User data	After
GtkCellRendererText			
edited	<Type here>	<Type here>	
GtkCellRenderer			

There is one last thing that we need to do: adjusting properties of adjustment1 that we created when setting properties of spin button. Close tree view editor and select adjustment1 from object tree. Now set minimum value to -5000, maximum value to 5000, step increment to 0.1, page increment to 10 and page size to 0 (only adjustments that are connected to scroll bar should have non-zero page size).



X	Y	L
3.000000	-2.000000	S
3.000000	-1.500000	
3.000000	-1.000000	
3.000000	-0.500000	Y
3.000000	0.000000	X
3.000000	0.500000	E

Adjustment Properties - GtkAdjustment [adjustme...

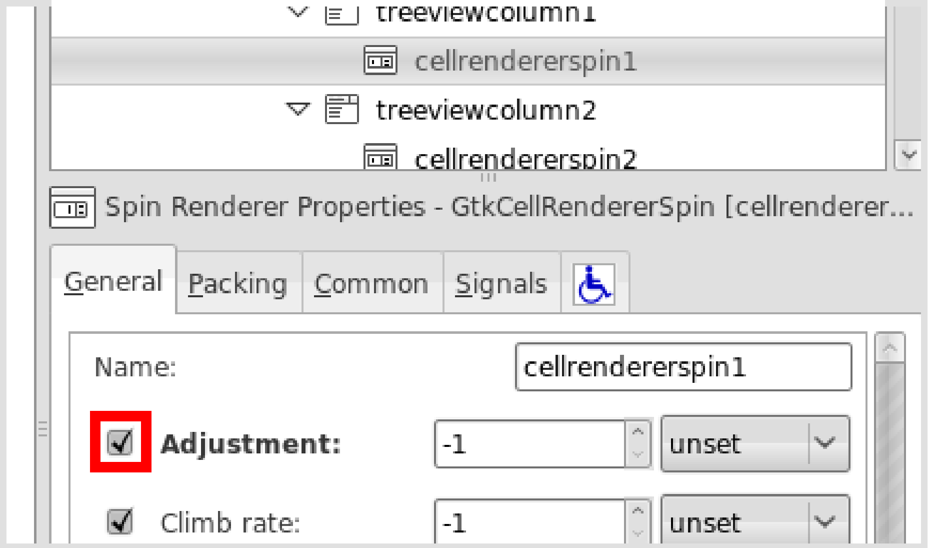
Name: adjustment1
Value: 0.00
Minimum Value: -5000.00
Maximum Value: 5000.00
Step Increment: 0.10
Page Increment: 10.00
Page Size: 0.00

You can see that values that we entered into list store are now displayed inside tree view. But we have a problem: labels are not visible because we set size request of the scrolled window too low. But how can we determine how much space will we need on different themes with different font sizes? The answer is: "We cannot.". And this is why we'll modify our GUI in next part of this tutorial and make it more flexible.

As usual, you can get the latest glade file [http://tadeboro.googlepages.com/tut02.glade] or watch screencast [http://www.screentoaster.com/watch/stU0tWQk1IR1xcQ1RbU15ZV1Fd/glade3_tutorial_creating_data_backend] .

Little Glade3-3.6.7 bug

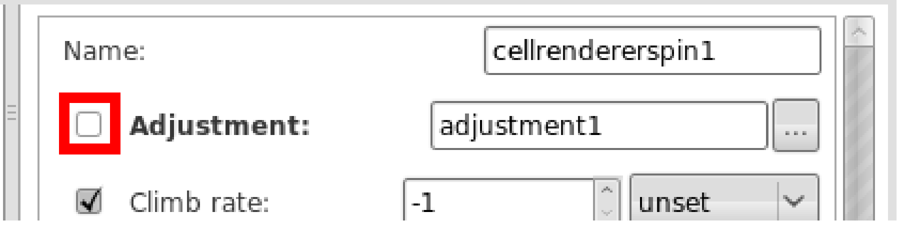
Glade3-3.6.7 has some problems with loading projects that use somewhat complex cell renderer settings. In our case, the problematic part is adjustment for spin buttons. You'll notice that when you load this project next time, adjustment property will be marked like it was set on per line basis (see image for details). All you have to do is remove those checks and you should be OK. This bug is already fixed in git version, so all that we have to do is wait for next release.



Name: cellrendererspin1

☑ Adjustment: -1 unset

☑ Climb rate: -1 unset



Name: cellrendererspin1

□ Adjustment: adjustment1 ...

☑ Climb rate: -1 unset

So long and stay healthy.

Posted 20th September 2009 by Tadej Borovšak

Labels: Glade, GTK+, GtkTreeModel, GtkTreeView, tutorial

18 View comments

Cenwen 20 September, 2009 20:50

Hello
Yet one and now yeah!! And i have print it and i'll see when i'll have the time.
I'm on the second part with a python file.
Thanks for this tutorials.

Reply

Syn 22 September, 2009 21:09

Thnx! I'm setting up a project for an exam and thanks to your tutorial it's growing better than i can. and i'm improving my skills, of course.

Reply

mark 24 September, 2009 03:04

Thanks a million ! I had no idea how to figure this out on my own ! I'm using glade 3.6.3 because of the horizontal vbox issue with 3.6.7... IT messed up my project when I tried to load iot with 3.6.7. I donno how to fix it though...

Reply

torkiano 25 September, 2009 00:51

@mark:

take a look at this bug: https://bugzilla.gnome.org/show_bug.cgi?id=590517 (is a bug in GTK+, not glade)

Reply

bootchk 15 October, 2009 11:55

Thanks. Well written. I know it took a lot of work.

Reply

Steve 31 December, 2009 03:14

Every time I try to edit the listdatastore (entering the coordinates), Glade crashes. Any advice?

Reply

Will 02 January, 2010 02:37

Steve,
Enter the data then press enter before trying to click on the next cell.

Reply

Doug 05 January, 2010 01:36

The spin buttons do not behave as I expect.
1. They work with integer values only, rounding the value taken from the model.
2. Single click does not change the value - I have to click and hold before the value changes.
3. The changed value is not retained after moving to another control - it reverts to the value loaded from tut.glade.

Are these behaviours controlled by other settable properties of the spin button control?

Otherwise a great tutorial - thanks.

Reply

Anonymous 22 March, 2010 20:44

What a contribution! Thanks!

Reply

rasmon 05 April, 2010 14:03

There are many useful stuff. Thanks. But I have some quastion. There are "tree model", "text buffers", "anjuta" (and some others) buttons in left view of your Glade application. This buttons is absent in my Glade. How I can install it in my Glade?
I have 4.3.2 version. I use Debian lenny.
Thanks in advance.

Reply

aguerreiro 06 April, 2010 14:40

Thanks, I'm trying to convert a hand-coded UI to Glade/GTKBuilder, and the TreeView stuff is tricky.

Reply

rasmon 07 April, 2010 17:03

sorry, mistake in my comment
"version 3.4.5"

Reply

Ivan Baldo 10 December, 2010 21:29

I have the same problem as Doug, anyone knows how to fix it?
Thanks!

Reply

Ivan Baldo 13 December, 2010 22:59

Remember that the cell renderers only SHOW a value and let you EDIT it but they DON'T UPDATE THE MODEL.
So, you should connect to the "edited" signal and set the new value to your GTK model.
Also, beware on locales that don't use the dot as decimal separator, see https://bugzilla.gnome.org/show_bug.cgi?id=637189 .
Hope this helps.

Reply

deciimal 18 December, 2010 16:58

There is a Catch.
I want to load data into it from a Table.
How would That be possible.

Reply

Tony Dardis 29 December, 2010 20:42

Thanks for a great tutorial!

I got up to "Now right-click newly created column and select "Add child Spin item" ..." and I'm stuck. It looks like my copy of Glade (3.6.7) running on Mac OS X has a problem with right-clicking on the Type field of the treeview column: a box with the word "Column" appears, but there's no pull-down selector (but if I repeatedly click, I can see that a pull-down is drawn, but then immediately goes away). Am I being stupid, or is there a problem?

Reply

Anonymous 06 June, 2011 04:08

You forgot to indicate that adjustement must be added from miscelaneous tab.

Hope it will be usefull to others.

Reply

Madan Mohan 22 August, 2013 15:41

how to access the row data, when I selected it, through Gtk, C

Reply

Enter your comment...

Comment as: Google Accou ▼

Publish Preview