



Workaround Swing Bugs

David Qiao, JIDE Software, Inc.

BOF-5133



Background of Myself and JIDE

- CTO and Founder of JIDE Software, Inc.
- Heavily use Swing in the last several years
- Build components and frameworks on top of Swing
 - New components
 - New features on top of existing Swing components
 - And to fix some Swing bugs

State of Swing Bugs

> Bugs

- Number of unresolved Swing bugs: 1839 (total 13204, or 14%)
- Number of unresolved AWT bugs: 1037 (total 11056, 9.4%)

> RFEs

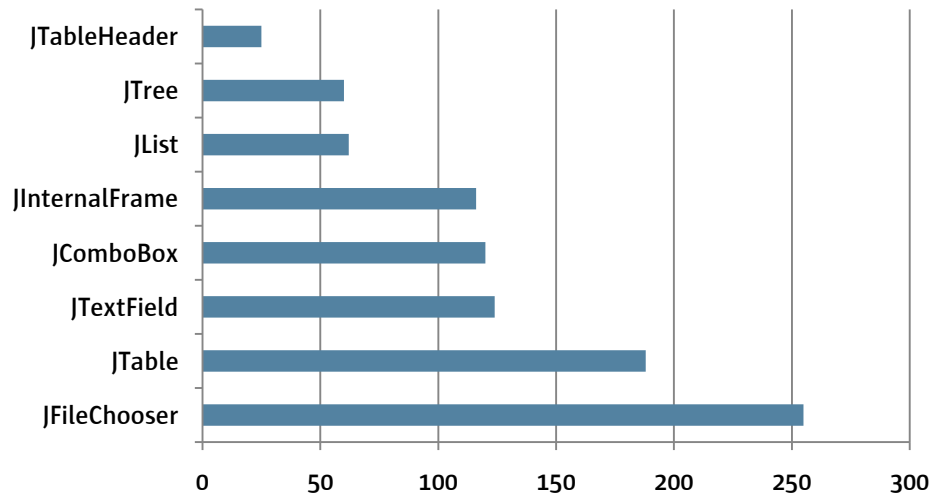
- Number of unresolved Swing RFEs: 568 (total 1892, 30%)
- Number of unresolved AWT RFEs: 298 (total 1185, 25%)

(as of April 21, 2008)

Where are Swing bugs

➤ Major components

- JFileChooser (255)
- JTable (188), JTextField (124), JComboBox (120),
- JInternalFrame (116), JList(62), JTree (60), JTableHeader (25)



(as of April 21, 2008)

Where are Swing bugs

➤ LookAndFeel

- GTK (165), Metal (101), Nimbus (33), Synth (23)
- Issues mimicking native OS – Windows XP and Vista

(as of April 21, 2008)

Some Famous pre-JDK5 Bugs

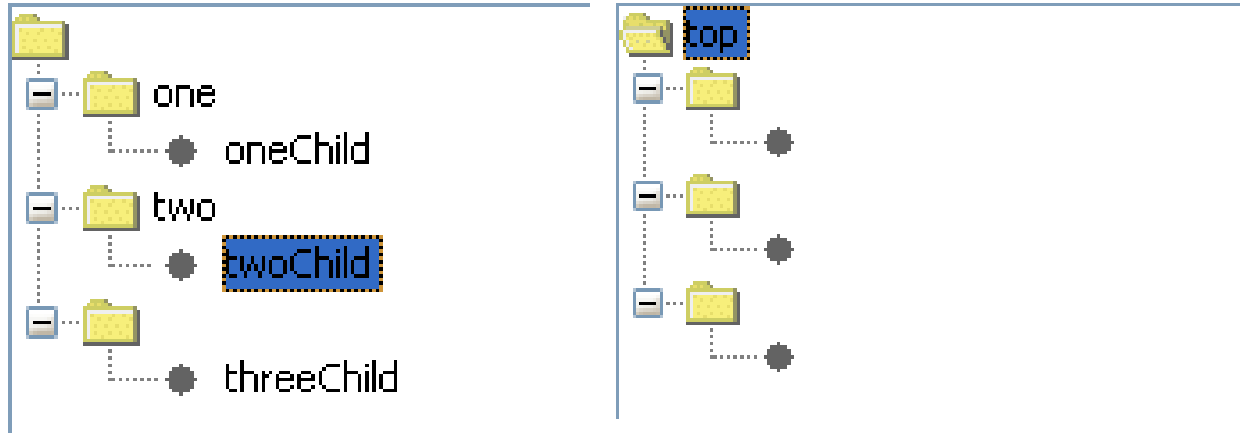
DEMO

TableHeader Height(ID # 4492484)

	column 1	column 2	column 3	column 4
0, 0	0, 1	0, 2	0, 3	0, 4
1, 0	1, 1	1, 2	1, 3	1, 4
2, 0	2, 1	2, 2	2, 3	2, 4
3, 0	3, 1	3, 2	3, 3	3, 4
4, 0	4, 1	4, 2	4, 3	4, 4
5, 0	5, 1	5, 2	5, 3	5, 4
6, 0	6, 1	6, 2	6, 3	6, 4
7, 0	7, 1	7, 2	7, 3	7, 4
8, 0	8, 1	8, 2	8, 3	8, 4
9, 0	9, 1	9, 2	9, 3	9, 4

	column 1	column 2	column 3	column 4
0, 0	0, 1	0, 2	0, 3	0, 4
1, 0	1, 1	1, 2	1, 3	1, 4
2, 0	2, 1	2, 2	2, 3	2, 4
3, 0	3, 1	3, 2	3, 3	3, 4
4, 0	4, 1	4, 2	4, 3	4, 4
5, 0	5, 1	5, 2	5, 3	5, 4
6, 0	6, 1	6, 2	6, 3	6, 4
7, 0	7, 1	7, 2	7, 3	7, 4
8, 0	8, 1	8, 2	8, 3	8, 4
9, 0	9, 1	9, 2	9, 3	9, 4

Disappearing JTree nodes (ID # 4373575)



Disappearing JTree nodes (Cont.)

- Got the Idea to make a new component – **StyledLabel**
- {Preview:bold} of {StyledLabel:f:red}
- **Preview of StyledLabel**
- Part of the open source JIDE Common Layer

StyledLabel Examples

Java™	CO ₂
Bold	<i>Italic</i>
Error	Background
Dotted	<u>Underlined</u>
Strikethrough	Double Strikethrough
Strikethrough and Underlined	<u>Customized Underlined</u>
Mixed Underlined Strikethrough Styles	Mixed Underlined Strikethrough (Disabled)

StyledLabel used as TreeCellRenderer

[-] **Severe Warnings (10 warnings)**

- [-] Error
- [-] Underlined
- [-] Strikethrough

[+] **Regular Warnings (2 warnings)**

[+] Messages (5 messages)

Famous Open Bugs

DEMO

Missing Table Selection Event

- When the mouse is pressed, the **valueIsAdjusting** flag is set to true. When the mouse is released, it is set to false.
- When a dialog pops up, there is mouse pressed event but no mouse released event
- **How to workaround it?**
- a. Don't check for **getValueIsAdjusting** flag if you are running behind the schedule
- b. Use another way to replace the missing event, such as using **FocusListener**.

Table Cell Selection (ID # 4138111)

`JTable#setCellSelectionEnabled(true)`

A	B	C	D	E	F	G	H	I	J
0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9
10,0	10,1	10,2	10,3	10,4	10,5	10,6	10,7	10,8	10,9
11,0	11,1	11,2	11,3	11,4	11,5	11,6	11,7	11,8	11,9
12,0	12,1	12,2	12,3	12,4	12,5	12,6	12,7	12,8	12,9
13,0	13,1	13,2	13,3	13,4	13,5	13,6	13,7	13,8	13,9
14,0	14,1	14,2	14,3	14,4	14,5	14,6	14,7	14,8	14,9
15,0	15,1	15,2	15,3	15,4	15,5	15,6	15,7	15,8	15,9
16,0	16,1	16,2	16,3	16,4	16,5	16,6	16,7	16,8	16,9
17,0	17,1	17,2	17,3	17,4	17,5	17,6	17,7	17,8	17,9
18,0	18,1	18,2	18,3	18,4	18,5	18,6	18,7	18,8	18,9
19,0	19,1	19,2	19,3	19,4	19,5	19,6	19,7	19,8	19,9

A	B	C	D	E	F	G	H	I	J
0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9
10,0	10,1	10,2	10,3	10,4	10,5	10,6	10,7	10,8	10,9
11,0	11,1	11,2	11,3	11,4	11,5	11,6	11,7	11,8	11,9
12,0	12,1	12,2	12,3	12,4	12,5	12,6	12,7	12,8	12,9
13,0	13,1	13,2	13,3	13,4	13,5	13,6	13,7	13,8	13,9
14,0	14,1	14,2	14,3	14,4	14,5	14,6	14,7	14,8	14,9
15,0	15,1	15,2	15,3	15,4	15,5	15,6	15,7	15,8	15,9
16,0	16,1	16,2	16,3	16,4	16,5	16,6	16,7	16,8	16,9
17,0	17,1	17,2	17,3	17,4	17,5	17,6	17,7	17,8	17,9
18,0	18,1	18,2	18,3	18,4	18,5	18,6	18,7	18,8	18,9
19,0	19,1	19,2	19,3	19,4	19,5	19,6	19,7	19,8	19,9

Table Cell Selection (Cont.)

- JTable has **TWO ListSelectionModels**, one for the rows and the other for the columns.
- If the table allows the row selection only, perfect
- If the table allows the column selection only, perfect
- If the table allows both row and column selection
 - Cells (1,1), (4, 4) are selected
 - ListSelectionModel for the rows (1 and 4)
 - ListSelectionModel for the columns (1 and 4)
 - It implies (4, 1) and (1, 4) are selected.

Table Cell Selection (Cont.)

- There is no easy workaround for this bug.
- Workaround becomes a new feature
- JIDE Grids introduces JideTable with non-contiguous mode to fix this bug.

A	B	C	D	E	F	G	H	I
0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8
4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8
8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8
9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8
10,0	10,1	10,2	10,3	10,4	10,5	10,6	10,7	10,8
11,0	11,1	11,2	11,3	11,4	11,5	11,6	11,7	11,8
12,0	12,1	12,2	12,3	12,4	12,5	12,6	12,7	12,8
13,0	13,1	13,2	13,3	13,4	13,5	13,6	13,7	13,8
14,0	14,1	14,2	14,3	14,4	14,5	14,6	14,7	14,8
15,0	15,1	15,2	15,3	15,4	15,5	15,6	15,7	15,8
16,0	16,1	16,2	16,3	16,4	16,5	16,6	16,7	16,8
17,0	17,1	17,2	17,3	17,4	17,5	17,6	17,7	17,8
18,0	18,1	18,2	18,3	18,4	18,5	18,6	18,7	18,8
19,0	19,1	19,2	19,3	19,4	19,5	19,6	19,7	19,8
20,0	20,1	20,2	20,3	20,4	20,5	20,6	20,7	20,8
21,0	21,1	21,2	21,3	21,4	21,5	21,6	21,7	21,8
22,0	22,1	22,2	22,3	22,4	22,5	22,6	22,7	22,8
23,0	23,1	23,2	23,3	23,4	23,5	23,6	23,7	23,8
24,0	24,1	24,2	24,3	24,4	24,5	24,6	24,7	24,8
25,0	25,1	25,2	25,3	25,4	25,5	25,6	25,7	25,8
26,0	26,1	26,2	26,3	26,4	26,5	26,6	26,7	26,8

TableSelectionModel: the interface

- void addSelection(int row, int column)
- void removeSelection(int row, int column)
- void setSelection(int row, int column)
- void clearSelection()
- boolean isSelected(int row, int column)
- void addTableSelectionListener(TableSelectionListener l)
- void removeTableSelectionListener(TableSelectionListener l)
- ...

DefaultTableSelectionModel: the implementation

- We still use ListSelectionModel and DefaultListSelectionModel internally
- Each column has its own ListSelectionModel

Putting Them Together

- **JideTable** is the place for all minor enhancements and bug fixes for **JTable**
- **Override two important methods**
 - `public void changeSelection(int rowIndex, int columnIndex, boolean toggle, boolean extend)`
 - `public boolean isCellSelected(int row, int column)`

How to workaround (or fix) Swing bugs

DEMO

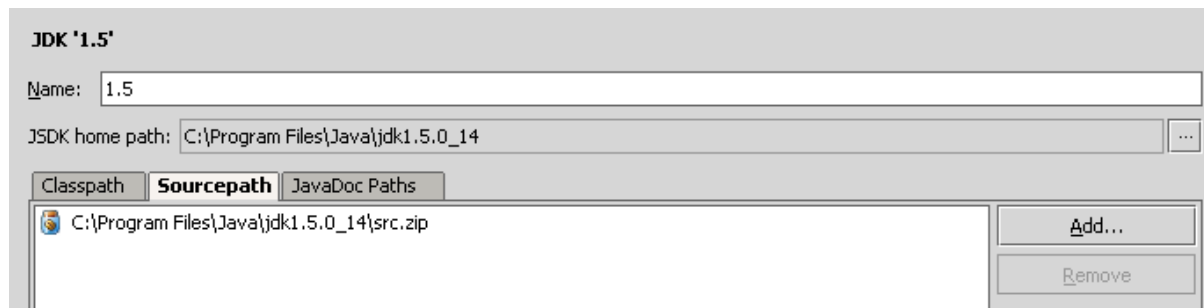
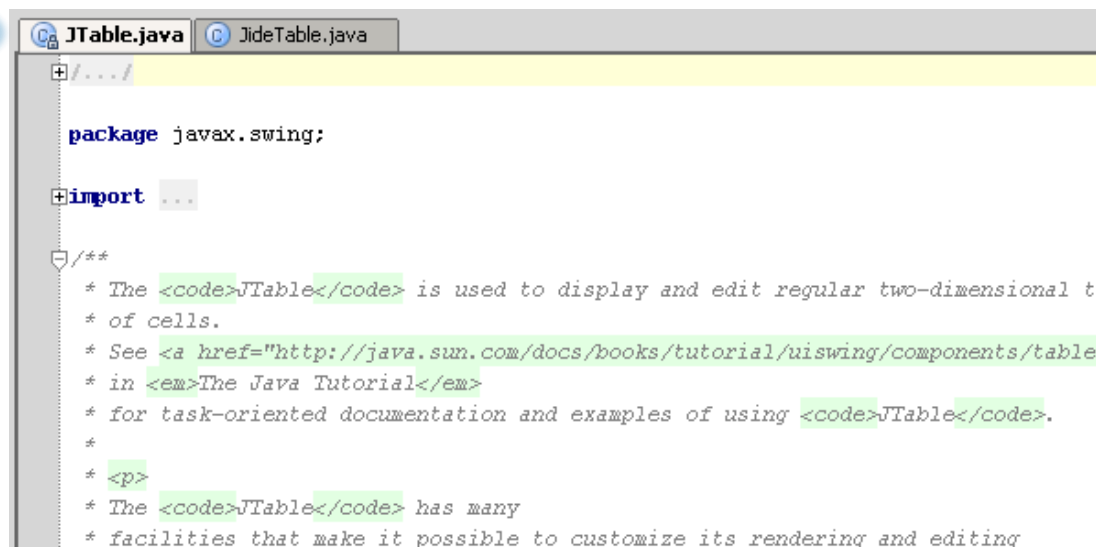
Do the Research First

- Search Sun Bug Database: <http://bugs.sun.com/>
 - Many bug reports have workarounds
- Search on the forums:
<http://forums.java.net/jive/category.jspa?categoryID=53>
- Search on the web

- **Do the research before you waste time on it**

Read JDK Source Code

- Make sure you link the JDK source code in your Java IDE.

```

package javax.swing;

import ...

/**
 * The JTable is used to display and edit regular two-dimensional ta
 * of cells.
 * See 
!\[\]\(6c9d205ef3bc86204ddce93eeadbb481\_img.jpg\)

```

Locate the Bug

➤ Swing is modularized

- **The Model** – most Swing component has model class – such as TableModel and TableColumnModel for JTable, ListModel and ListSelectionModel in JList, etc.
- **The CellRenderer and CellEditor** (in the case of JTable, JList, JTree)
- **The Component** – such as JTable, JList etc.
- **The ComponentUI class** – such as TableUI – usually one for each L&F

Provide Your Own Implementation

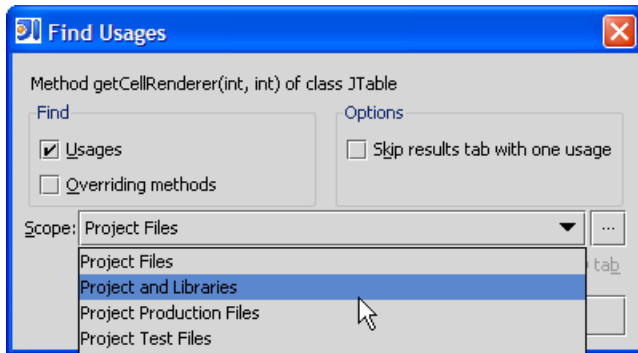
- If the default implementation has bug
 - Implement your own and replace the existing one
 - Apply to bugs in the model, the cell renderer and editor
- The table selection model example just mentioned

Override Methods

- To prevent calling to the super method
 - Override and do nothing
- To customize the behavior of the super method
 - Call `super.method()`, then add your own code
- To change the condition before calling the super method
 - Add your own code then call `super.method()`
- To provide an alternative implement of the super method
 - Copy the code from `super.method` and change it

Override Methods (Cont.)

- Use Java IDE “find usages” feature to find out who are using this method (including the usages in JDK itself) so that you know the scope the affected area after overriding the method



- Call super if you can and provide your fix before or after calling super.
- Comment clearly if you have to duplicate JDK code

Provide your own ComponentUI class

- The most powerful and extensive way to get what you want.
- However, avoid it if you can
 - Unless you have your own L&F and you only plan to support this L&F for you application.
 - Difficult to support all L&Fs
 - Difficult to maintain for different JDK's

A Real Example – Disabling a JTable

- There is no visual difference when `table.setEnabled(false)` is called

THANK YOU



David Qiao, JIDE Software, Inc.
Workaround Swing Bugs

BOF-5133

