# Java Code Examples for javax.swing.table.TableModel

The following are top voted examples for showing how to use javax.swing.table.TableModel. These examples are extracted from open source projects. You can vote up the examples you like and your votes will be used in our system to generate more good examples.

**Popular Methods**

- getRowCount ( )
- getValueAt ( )
- getColumnCount ( )
- addTableModelListener ( )
- setValueAt ( )
- getColumnName ( )
- removeTableModelListener ( )

**Related Classes**

- java.io.File
- java.util.Arrays
- java.util.Date
- java.awt.Color
- java.awt.event.ActionEvent
- java.util.Vector
- java.awt.event.ActionListener
- javax.swing.JFrame
- javax.swing.JPanel
- java.awt.Dimension
- javax.swing.JLabel
- javax.swing.JOptionPane
- javax.swing.JButton
- java.awt.Font
- java.awt.BorderLayout
- java.awt.event.MouseEvent
- javax.swing.JTextField
- javax.swing.JScrollPane
- java.awt.Component
- javax.swing.ImageIcon
- javax.swing.SwingUtilities
- java.awt.Point
- java.awt.event.MouseAdapter
- javax.swing.JComponent
- javax.swing.BorderFactory

**Example 1**

Project: *marathonv5*   File: *JTableJavaElement.java*   *Source Code and License*        8 votes

```java
private void validate(int viewRow, int viewCol) {
    JTable table = (JTable) getComponent();
    try {
        int row = table.convertRowIndexToModel(viewRow);
        int col = table.convertColumnIndexToModel(viewCol);
        TableModel model = table.getModel();
        if (row >= 0 && row < model.getRowCount() && col >= 0 && col < model.getColumnCount()) {
            if (table.isCellEditable(viewRow, viewCol)) {
                return;
            } else {
                throw new NoSuchElementException("The cell is not editable on JTable: (" + viewRow
+ ", " + viewCol + ")",
                        null);
            }
        }
    } catch (IndexOutOfBoundsException e) {
    }
    throw new NoSuchElementException("Invalid row/col for JTable: (" + viewRow + ", " + viewCol +
")", null);
}
```

**Example 2**

Project: *KernelHive*   File: *SourceFilePropertiesDialog.java*   *Source Code and License*        6 votes

```java
private boolean saveProperties() {
        final Map<String, Object> properties = new HashMap<String, Object>();
        final TableModel model = table.getModel();
        if (model.getColumnCount() == 2) {
                for (int i = 0; i < model.getRowCount(); i++) {
                        final String key = (String) model.getValueAt(i, 0);
                        final Object value = model.getValueAt(i, 1);
                        if (properties.containsKey(key)) {
                                return false;
                        } else {
                                properties.put(key, value);
                        }
                }
                sourceFile.setProperties(properties);
                return true;
        } else {
                return false;
        }
}
```

**Example 3**

Project: *ramus*   File: *TableEditorTable.java*   *Source Code and License*        6 votes

```java
@Override
public void setModel(TableModel dataModel) {
    super.setModel(dataModel);
    if (!(dataModel instanceof TableEditorModel))
        return;
    TableEditorModel model = (TableEditorModel) dataModel;
    Engine engine = framework.getEngine();
    AccessRules rules = framework.getAccessRules();
    for (int i = 0; i < plugins.length; i++) {
        AttributePlugin plugin = framework.findAttributePlugin(attributes
                .get(i));
```

```java
        plugins[i] = plugin;
        cellEditors[i] = plugin.getTableCellEditor(engine, rules,
                attributes.get(i));

        if (cellEditors[i] == null) {
            cellEditors[i] = new DialogedTableCellEditor(engine, rules,
                    attributes.get(i), plugins[i], framework);
            model.setSaveValue(i, false);
        }

        cellRenderers[i] = plugin.getTableCellRenderer(engine, rules,
                attributes.get(i));
    }
}
```

**Example 4**

```java
public static JTable createRawTable(TableModel model)
{
        JTable table = new JTable(model)
        {
                @Override
                public boolean getScrollableTracksViewportHeight()
                {
                        // fetch the table's parent
                        Container viewport = getParent();

                        // if the parent is not a viewport, calling this isn't useful
                        if( !(viewport instanceof JViewport) )
                        {
                                return false;
                        }

                        // return true if the table's preferred height is smaller
                        // than the viewport height, else false
                        return getPreferredSize().height < viewport.getHeight();
                }
        };
        return table;
}
```

**Example 5**

```java
public void setTableModel(TableModel model) {
    if (model == null) {
        table.clearSelection();
        table.setModel(new DefaultTableModel());
        removeAll();
        add(noDataContainer, BorderLayout.CENTER);
    } else {
        int selectedRow = getSelectedRow();
        table.setModel(model);
        if (selectedRow != -1)
            table.getSelectionModel().setSelectionInterval(selectedRow,
                                                            selectedRow);
        removeAll();
        add(dataContainer, BorderLayout.CENTER);
    }

    validate();
    repaint();
}
```

**Example 6**

```java
public Vector<String> getAllShortcutsForAction(String actionName) {
    System.out.println("[TEST_DEBUG]");
    System.out.println("[TEST_DEBUG] ### Examining all shortcuts for action: " + actionName);
    Vector<String> lstr = new Vector<String>();
```

```
        String tmpStr = actionSearchByName().getText();
        searchActionName(actionName);
        TableModel tm = getActionsTableModel();
        String _str;
        String _scStr;
        for (int i = 0; i < tm.getRowCount(); i++) {
            _str = tm.getValueAt(i, 0).toString();
            if (_str.toLowerCase().equals(actionName.toLowerCase()) ||
_str.toLowerCase().equals(actionName.toLowerCase() + " (alternative shortcut)")) {
                _scStr = tm.getValueAt(i, 1).toString().toLowerCase();
                lstr.add(_scStr);
                System.out.println("[TEST_DEBUG]  -> found action \"" + _str + "\" with shortcut " +
_scStr);
            }
        }
        searchActionName(tmpStr);
        System.out.println("[TEST_DEBUG] ### Examining all shortcuts done");
        return lstr;
}
```

**Example 7**

Project: *BadIntent*  File: *ParcelPanel.java*    *Source Code and License*          6 votes

```
public List<ParcelOperationDAO> retrievedParcelOperations(){
    hasBeenEdited = false;
    List<ParcelOperationDAO> ops = new LinkedList<>();
    TableModel model = table.getModel();
    if (model.getRowCount() != parcelOperations.size()) {
        hasBeenEdited = true;
    }
    for (int i = 0; i < model.getRowCount(); i++){
        String parcelOperation = (String) model.getValueAt(i, 0);
        Object parcelValue = model.getValueAt(i, 1);
        String originalOperation = parcelOperations.get(i).parcelType;
        Object originalValue = parcelOperations.get(i).value;
        if (parcelValue == null && originalValue == null) {
            ops.add(parcelOperations.get(i));
        } else if (parcelValue == null || originalValue == null) {
            hasBeenEdited = true;
            ops.add(new ParcelOperationDAO(parcelOperation, parcelValue));
        } else if (!originalOperation.equals(parcelOperation) ||
!originalValue.equals(parcelValue)) {
            hasBeenEdited = true;
            ops.add(new ParcelOperationDAO(parcelOperation, parcelValue));
        } else {
            ops.add(parcelOperations.get(i));
        }
    }
    return ops;
}
```

**Example 8**

Project: *OpenDiabetes*  File: *GridSwing.java*    *Source Code and License*          6 votes

```
public static void autoSizeTableColumns(JTable table) {

        TableModel  model         = table.getModel();
        TableColumn column        = null;
        Component   comp          = null;
        int         headerWidth   = 0;
        int         maxCellWidth = Integer.MIN_VALUE;
        int         cellWidth    = 0;
        TableCellRenderer headerRenderer =
            table.getTableHeader().getDefaultRenderer();

        for (int i = 0; i < table.getColumnCount(); i++) {
            column = table.getColumnModel().getColumn(i);
            comp = headerRenderer.getTableCellRendererComponent(table,
                    column.getHeaderValue(), false, false, 0, 0);
            headerWidth  = comp.getPreferredSize().width + 10;
            maxCellWidth = Integer.MIN_VALUE;

            for (int j = 0; j < Math.min(model.getRowCount(), 30); j++) {
                TableCellRenderer r = table.getCellRenderer(j, i);
```

```
                comp = r.getTableCellRendererComponent(table,
                                            model.getValueAt(j, i),
                                            false, false, j, i);
                cellWidth = comp.getPreferredSize().width;

                if (cellWidth >= maxCellWidth) {
                    maxCellWidth = cellWidth;
                }
            }

            column.setPreferredWidth(Math.max(headerWidth, maxCellWidth)
                                    + 10);
        }
    }
```

**Example 9**

```
private void doExport(TableModel model, String title, Writer writer,
                        TracerProgressObject progress) throws IOException {
    progress.setText("Initializing export...");
    writeHeader(model, title, writer, progress);

    if (progress.isFinished()) return;

    progress.addStep("Exporting data...");
    writeData(model, title, writer, progress);

    if (progress.isFinished()) return;

    progress.setText("Finishing export...");
    writeFooter(model, title, writer, progress);

    if (progress.isFinished()) return;

    progress.setText("Data exported");
    progress.finish();
}
```

**Example 10**

```
/**
 * Invoked when an action occurs.
 */
public void actionPerformed(final ActionEvent e) {
    List<JavaPlatform> platforms = JavaPlatform.getPlatforms();
    String[] columnNames = new String[] { Bundle.LBL_JavaPlatform(),
                                    Bundle.LBL_LastCalibrated() };
    Object[][] columnData = new Object[platforms.size()][2];
    for (int i = 0; i < platforms.size(); i++)
        columnData[i] = new Object[] { platforms.get(i), null };
    final TableModel model = new DefaultTableModel(columnData, columnNames) {
        public boolean isCellEditable(int row, int column) { return false; }
    };

    SwingUtilities.invokeLater(new Runnable() {
        public void run() { displayUI(model); }
    });

    RequestProcessor.getDefault().post(new Runnable() {
        public void run() { refreshTimes(model); }
    });
}
```

**Example 11**

```
private void okButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_okButtonActionPerformed
        // Update new enabled/disabled status for the Detectors
        int num = factoryList.size();
```

```java
        DefaultSortedTableModel sorter = (DefaultSortedTableModel) detectorTable.getModel();
        TableModel model = sorter.getBaseTableModel();
        for (int i = 0; i < num; ++i) {
                DetectorFactory factory = factoryList.get(i);
                Boolean enabled = (Boolean) model.getValueAt(i, ENABLED_COLUMN);
                factory.setEnabled(enabled.booleanValue());
        }
        closeDialog();
}
```

**Example 12**

```java
private static TableModel createModel(final Map<String,String> callbacks) {
    final Object[][] data = new Object[callbacks.size()][];
    final Iterator<Map.Entry<String,String>> it = callbacks.entrySet().iterator();
    for (int i=0; it.hasNext(); i++) {
        final Map.Entry<String,String> entry = it.next();
        data[i] = new Object[] {entry.getKey(),entry.getValue()};
    }
    return new DefaultTableModel(
        data,
        new String[] {
            NbBundle.getMessage(JFXDownloadModePanel.class,
"JFXJavaScriptCallbacksPanel.tableCallbacks.columnModel.title0"), //NOI18N
            NbBundle.getMessage(JFXDownloadModePanel.class,
"JFXJavaScriptCallbacksPanel.tableCallbacks.columnModel.title1") //NOI18N
        }) {
        @Override
        public boolean isCellEditable(int row, int column) {
            return column != 0;
        }

        @Override
        public Class<?> getColumnClass(int columnIndex) {
            switch (columnIndex) {
                case 0:
                    return String.class;
                case 1:
                    return String.class;
                default:
                    throw new IllegalStateException();
            }
        }

    };
}
```

**Example 13**

```java
public void selectWithSpecialCharsIHeader() throws Throwable {
    final JTable table = (JTable) ComponentUtils.findComponent(JTable.class, frame);
    final LoggingRecorder lr = new LoggingRecorder();
    siw(new Runnable() {
        @Override public void run() {
            JTableHeader tableHeader = table.getTableHeader();
            String[] columnNames = new String[table.getColumnCount()];
            for (int i = 0; i < table.getColumnCount(); i++) {
                columnNames[i] = table.getColumnName(i);
            }
            columnNames[3] = " # of \\Years[0,20]";
            TableModel model = table.getModel();
            try {
                Field field = model.getClass().getDeclaredField("columnNames");
                field.setAccessible(true);
                field.set(model, columnNames);
            } catch (Exception e) {
                e.printStackTrace();
            }
            table.setModel(model);
            tableHeader.getColumnModel().getColumn(3).setHeaderValue(" # of \\Years[0,20]");
            table.invalidate();
            table.setRowSelectionAllowed(true);
```

```
table.getColumnModel().getSelectionModel().setSelectionMode(ListSelectionModel.MULTIPLE_INTERVAL_S
ELECTION);

table.getSelectionModel().setSelectionMode(ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);
            table.addRowSelectionInterval(1, 1);
            table.addRowSelectionInterval(3, 3);
            table.addColumnSelectionInterval(3, 3);
            table.addColumnSelectionInterval(2, 2);
            RTable rTable = new RTable(table, null, null, lr);
            rTable.focusLost(null);
        }
    });
    Call call = lr.getCall();
    AssertJUnit.assertEquals("select", call.getFunction());
    AssertJUnit.assertEquals("rows:[1,3],columns:[Sport, ## of \\Years[0#;20]]", call.getState());
}
```

**Example 14**

```java
@Override
public void store() {
    Properties p = Properties.getDefault().getProperties("debugger.options.JPDA");
    p.setBoolean("UseStepFilters", useStepFiltersCheckBox.isSelected());
    p.setBoolean("FilterSyntheticMethods", filterSyntheticCheckBox.isSelected());
    p.setBoolean("FilterStaticInitializers", filterStaticInitCheckBox.isSelected());
    p.setBoolean("FilterConstructors", filterConstructorsCheckBox.isSelected());
    TableModel filterClassesModel = filterClassesTable.getModel();
    Set<String> allFilters = new LinkedHashSet<String>();
    Set<String> enabledFilters = new HashSet<String>();
    int n = filterClassesModel.getRowCount();
    for (int i = 0; i < n; i++) {
        boolean isEnabled = (Boolean) filterClassesModel.getValueAt(i, 0);
        String clazz = (String) filterClassesModel.getValueAt(i, 1);
        allFilters.add(clazz);
        if (isEnabled) {
            enabledFilters.add(clazz);
        }
    }
    Properties.getDefault ().getProperties ("debugger").
            getProperties ("sources").getProperties ("class_filters").
            setCollection (
                "all",
                allFilters
            );
    Properties.getDefault ().getProperties ("debugger").
            getProperties ("sources").getProperties ("class_filters").
            setCollection (
                "enabled",
                enabledFilters
            );
    p.setBoolean("StepThroughFilters", stepThroughFiltersCheckBox.isSelected());
}
```

**Example 15**

```java
public void store(VariablesFormatter f) {
    f.setName(nameTextField.getText());
    f.setClassTypes(classTypesTextField.getText());
    f.setIncludeSubTypes(subtypesCheckBox.isSelected());
    f.setValueFormatCode(valueFormatCheckBox.isSelected() ? valueEditorPane.getText() : "");
    f.setChildrenFormatCode(childrenCodeEditorPane.getText());
    TableModel tableModel = childrenVariablesTable.getModel();
    f.getChildrenVariables().clear();
    for (int i = 0; i < tableModel.getRowCount(); i++) {
        f.addChildrenVariable((String) tableModel.getValueAt(i, 0), (String)
tableModel.getValueAt(i, 1));
    }
    f.setUseChildrenVariables(childrenVariablesRadioButton.isSelected());
    f.setChildrenExpandTestCode(testChildrenCheckBox.isSelected() ?
testChildrenEditorPane.getText() : "");
}
```

**Example 16**

```java
public String[] getIncludedNamespaces() {
    List<String> incs = new ArrayList<String>();
    TableModel model = includesTable.getModel();

    for(int i = 0; i < model.getRowCount(); i++) {
        Boolean selected = (Boolean) model.getValueAt(i, 0);
        if(selected != null && selected == Boolean.TRUE) {
            String namespace = (String) model.getValueAt(i, 1);
            incs.add(namespace);
        }
    }
    return incs.toArray(new String[0]);
}
```

**Example 17**

```java
public void setTableModel(TableModel tableModel) {
        if (this.tableModel != null) {
                this.tableModel.removeTableModelListener(tableModelListener);
        }

        this.tableModel = tableModel;
        if (this.tableModel != null) {
                this.tableModel.addTableModelListener(tableModelListener);
        }

        clearSortingState();
        fireTableStructureChanged();
}
```

**Example 18**

```java
public void setTableModel(TableModel tableModel) {
    if (this.tableModel != null) {
        this.tableModel.removeTableModelListener(tableModelListener);
    }
    this.tableModel = tableModel;
    if (this.tableModel != null) {
        this.tableModel.addTableModelListener(tableModelListener);
    }
    clearSortingState();
    fireTableStructureChanged();
}
```

**Example 19**

```java
public static void autoSizeTableColumns(JTable table) {

        TableModel  model       = table.getModel();
        TableColumn column      = null;
        Component   comp        = null;
        int         headerWidth = 0;
        int         maxCellWidth = Integer.MIN_VALUE;
        int         cellWidth   = 0;
        TableCellRenderer headerRenderer =
            table.getTableHeader().getDefaultRenderer();

        for (int i = 0; i < table.getColumnCount(); i++) {
            column = table.getColumnModel().getColumn(i);
            comp = headerRenderer.getTableCellRendererComponent(table,
                    column.getHeaderValue(), false, false, 0, 0);
            headerWidth  = comp.getPreferredSize().width + 10;
            maxCellWidth = Integer.MIN_VALUE;
```

```
                for (int j = 0; j < Math.min(model.getRowCount(), 30); j++) {
                    TableCellRenderer r = table.getCellRenderer(j, i);

                    comp = r.getTableCellRendererComponent(table,
                                                        model.getValueAt(j, i),
                                                        false, false, j, i);
                    cellWidth = comp.getPreferredSize().width;

                    if (cellWidth >= maxCellWidth) {
                        maxCellWidth = cellWidth;
                    }
                }

                column.setPreferredWidth(Math.max(headerWidth, maxCellWidth)
                                        + 10);
            }
        }
```

**Example 20**

```
private static void checkTableGridLines() {

        TableModel dataModel = new AbstractTableModel() {
            public int getColumnCount() {
                return 10;
            }

            public int getRowCount() {
                return 10;
            }

            public Object getValueAt(int row, int col) {
                return " ";
            }
        };

        DefaultTableCellRenderer r = new DefaultTableCellRenderer();
        r.setOpaque(true);
        r.setBackground(CELL_RENDERER_BACKGROUND_COLOR);

        JTable table = new JTable(dataModel);
        table.setSize(WIDTH, HEIGHT);
        table.setDefaultRenderer(Object.class, r);
        table.setGridColor(GRID_COLOR);
        table.setShowGrid(true);
        table.setShowHorizontalLines(true);
        table.setShowVerticalLines(true);
        table.setBackground(TABLE_BACKGROUND_COLOR);

        checkTableGridLines(table);
    }
```

**Example 21**

```
/** Throws an UnsupportedOperationException when called by user code.  Replacing
 *  the data model of property sheets is unsupported.  You can change the model
 *  that determines what properties are shown - see <code>setPropertySetModel()</code>. */
@Override
public void setModel(TableModel model) {
    if (initialized) {
        throw new UnsupportedOperationException(
            "Changing the model of a property sheet table is not supported.  If you want to change
the set of properties, ordering or other characteristings, see setPropertySetModel()."
        ); //NOI18N
    }

    super.setModel(model);
}
```

**Example 22**

```java
public TableModel getTableModelFor(Object selectedNode) {
    if (selectedNode instanceof DataModel) {
        DataModel scenario = (DataModel) selectedNode;
        scenario.loadTableModel();
        return scenario;
    }
    return new DefaultTableModel();
}
```

**Example 23**

```java
public TableModelComparator(TableModel model) {
    this.model = model;

    // XXX - Should actually listen for column changes and resize
    columns = new int[model.getColumnCount()];
    columns[0] = -1;
}
```

**Example 24**

```java
@Override
public Reportable createReportable(Object renderable, IOContainer ioContainer) {
        TableModel tableModel = getTableModel(renderable, ioContainer, true);
        if (tableModel != null) {
                return new DefaultTableable(tableModel, this);
        }
        return null;
}
```

**Example 25**

```java
public Map listAbbreviations() {
    TableModel model = getTemplateTable().getModel();
    int rowCount = model.getRowCount();
    Map result = new HashMap();

    for (int cntr = 0; cntr < rowCount; cntr++) {
        result.put((String) model.getValueAt(cntr, 0), (String) model.getValueAt(cntr, 1));
    }

    return result;
}
```

**Example 26**

```java
public void setTableModel(TableModel tableModel) {

        if (this.tableModel != null) {
            this.tableModel.removeTableModelListener(tableModelListener);
        }

        this.tableModel = tableModel;

        if (this.tableModel != null) {
            this.tableModel.addTableModelListener(tableModelListener);
        }

        clearSortingState();
        fireTableStructureChanged();
    }
```

**Example 27**

```java
@Override
public boolean include(RowFilter.Entry<? extends TableModel,? extends Integer> entry)  {
    for (int colIdx : cols) {
        Object val = entry.getValue(colIdx);
        if (testValue(val)) {
                return true;
            }
        }
    return false;
}
```

**Example 28**

```java
private static int getIndexByName(TableModel model, String projectName) {
    int length = model.getRowCount();
    String name;
    for(int i =0; i< length; i++) {
        name = (String)model.getValueAt(i, 1);
        if(name.startsWith(projectName)) return i;
    }
    return 0;
}
```

**Example 29**

```java
public static void autoSizeTableColumns(JTable table) {

        TableModel  model        = table.getModel();
        TableColumn column       = null;
        Component   comp         = null;
        int         headerWidth  = 0;
        int         maxCellWidth = Integer.MIN_VALUE;
        int         cellWidth    = 0;
        TableCellRenderer headerRenderer =
            table.getTableHeader().getDefaultRenderer();

        for (int i = 0; i < table.getColumnCount(); i++) {
            column = table.getColumnModel().getColumn(i);
            comp = headerRenderer.getTableCellRendererComponent(table,
                    column.getHeaderValue(), false, false, 0, 0);
            headerWidth  = comp.getPreferredSize().width + 10;
            maxCellWidth = Integer.MIN_VALUE;

            for (int j = 0; j < Math.min(model.getRowCount(), 30); j++) {
                TableCellRenderer r = table.getCellRenderer(j, i);

                comp = r.getTableCellRendererComponent(table,
                                                  model.getValueAt(j, i),
                                                  false, false, j, i);
                cellWidth = comp.getPreferredSize().width;

                if (cellWidth >= maxCellWidth) {
                    maxCellWidth = cellWidth;
                }
            }

            column.setPreferredWidth(Math.max(headerWidth, maxCellWidth)
                            + 10);
        }
    }
```

**Example 30**

```java
public InfoEntry(int column, TableModel model, String[] cmd) {
        this.cmd = cmd;
        this.column = column;
        this.model = model;
        label = new JLabel();
        label.setBackground(Color.white);
        label.setForeground(Color.blue);
}
```

**Example 31**

```java
public TableBase(TableModel m, TableCellRenderer renderer, TransferHandler transfer, int start,
int end)
{
        super(m);

        startModelColumn = start;
        endModelColumn = end;
        setDefaultRenderer(Object.class, renderer);
        setTransferHandler(transfer);
        setAutoCreateColumnsFromModel( false );
        addMouseListener(new DClickWatch());
        getTableHeader().setReorderingAllowed(false);
        setCellSelectionEnabled(true);
        setSelectionMode(ListSelectionModel.SINGLE_INTERVAL_SELECTION);
        setRowHeight(36);
}
```

**Example 32**

```java
protected void writeData(TableModel model, String title, Writer writer,
                        TracerProgressObject progress) throws IOException {
    int columnsCount = model.getColumnCount();
    int rowsCount = model.getRowCount();

    writeLine(writer, "  <TableData NumRows=\"" + rowsCount + // NOI18N
                        "\" NumColumns=\"" + columnsCount + "\">"); // NOI18N

    writeLine(writer, "    <TableHeader>"); // NOI18N
    for (int c = 0; c < columnsCount; c++)
        writeLine(writer, "      <TableColumn>" + model.getColumnName(c) + "</TableColumn>"); //
NOI18N
    writeLine(writer, "    </TableHeader>"); // NOI18N

    writeLine(writer, "    <TableBody>"); // NOI18N
    for (int r = 0; r < rowsCount; r++) {
        writeLine(writer, "      <TableRow>"); // NOI18N
        for (int c = 0; c < columnsCount; c++)
            writeLine(writer, "        <TableColumn>" + model.getValueAt(r, c) + "
</TableColumn>"); // NOI18N
        writeLine(writer, "      </TableRow>"); // NOI18N

        if (progress.isFinished()) break;

        if (step == 1) {
            progress.addStep();
        } else {
            int currentStep = (int)(r * step);
            if (currentStep > lastStep) {
                progress.addStep();
                lastStep = currentStep;
            }
        }
    }
    writeLine(writer, "    </TableBody>"); // NOI18N

    writeLine(writer, "  </TableData>"); // NOI18N
}
```

**Example 33**

```java
@Override
public void actionPerformed(ActionEvent ae) {

        @SuppressWarnings("unchecked")
        TableRowSorter<TableModel>sorter = (TableRowSorter<TableModel>)
getJTableThreadProtocolVector().getRowSorter();

        if (ae.getSource()==this.getJRadioButtonNoFilter()) {
                // --- Remove Filter ----------------
                sorter.setRowFilter(null);

        } else if (ae.getSource()==this.getJRadioButtonFilterAgents()) {
                // --- Set Filter -------------------
                RowFilter<Object,Object> agentFilter = new RowFilter<Object, Object>() {

                        public boolean include(Entry<? extends Object, ? extends Object> entry)
{

                                // --- get column with ThreadDetail-Instance (ThreadName) ---
                                if(entry.getValue(1) instanceof ThreadDetail) {
                                        ThreadDetail tt = (ThreadDetail)entry.getValue(1);
                                        if(tt.isAgent() == true) {
                                                return true;
                                        }
                                }
                                return false;
                        }
                };
                sorter.setRowFilter(agentFilter);
        }
}
```

**Example 34**

```java
protected void writeHeader(TableModel model, String title, Writer writer,
                          TracerProgressObject progress) throws IOException {
    writeLine(writer, "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\"
\"http://www.w3.org/TR/html4/loose.dtd\">"); // NOI18N
    writeLine(writer);

    writeLine(writer, "<html>"); // NOI18N
    writeLine(writer, "<head>"); // NOI18N
    writeLine(writer, "  <title>"); // NOI18N
    writeLine(writer, "    " + title); // NOI18N
    writeLine(writer, "  </title>"); // NOI18N
    writeLine(writer, "</head>"); // NOI18N
    writeLine(writer);

    writeLine(writer, "<body>"); // NOI18N
}
```

**Example 35**

```java
@Override
public TableModel getTableModel(Object renderable, IOContainer ioContainer, boolean isReporting)
{
        LinearModel linearModel = (LinearModel) renderable;
        DataTable weightDataTable = linearModel.createWeightsTable();
        if (weightDataTable != null) {
                return new DataTableViewerTableModel(weightDataTable);
        } else {
                return null;
        }
}
```

**Example 36**

```
/**
 * reverts the selected state of all the detectors to their defaults as specified in the
 findbugs.xml file
 *
 * @param evt the swing event corresponding to the mouse click of the Restore Defaults button
 */
private void restoreDefaultsButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_restoreDefaultsButtonActionPerformed
        Iterator<DetectorFactory> i = DetectorFactoryCollection.instance().factoryIterator();
        DefaultSortedTableModel sorter = (DefaultSortedTableModel) detectorTable.getModel();
        TableModel model = sorter.getBaseTableModel();
        int row = 0;
        while (i.hasNext()) {
                DetectorFactory factory = i.next();
                model.setValueAt(factory.isDefaultEnabled() ? Boolean.TRUE : Boolean.FALSE, row++,
ENABLED_COLUMN);
        }
}
```

**Example 37**

Project: *incubator-netbeans*  File: *PropertyEditorPanel.java*   *Source Code and License*        5 votes

```
public PropertyEditorPanel(Properties initalValue, boolean editable) {
    initComponents();
    this.value = initalValue;
    this.editable = editable;
    propertyTable.putClientProperty(
            "terminateEditOnFocusLost", Boolean.TRUE);                //NOI18N
    updateTableFromEditor();
    final TableModel tm = propertyTable.getModel();
    tm.addTableModelListener(new TableModelListener() {
        @Override
        public void tableChanged(TableModelEvent tme) {
            synchronized (PropertyEditorPanel.this) {
                if (updateing) {
                    return;
                }
                updateing = true;
                Properties p = new Properties();
                for (int i = 0; i < tm.getRowCount(); i++) {
                    p.setProperty((String) tm.getValueAt(i, 0), (String) tm.getValueAt(i, 1));
                }
                Properties oldValue = value;
                value = p;
                firePropertyChange(PROP_VALUE, oldValue, value);
                updateing = false;
            }
        }
    });
    propertyTable.getSelectionModel().addListSelectionListener(
            new ListSelectionListener() {
                @Override
                public void valueChanged(ListSelectionEvent lse) {
                    updateRemoveButtonSensible();
                }
            });
    updateAddButtonSensible();
    updateRemoveButtonSensible();
}
```

**Example 38**

Project: *MaxSim*  File: *TableModelComparator.java*    *Source Code and License*         5 votes

```
public TableModelComparator(TableModel model) {
    this.model = model;

    // XXX - Should actually listen for column changes and resize
    columns = new int[model.getColumnCount()];
    columns[0] = -1;
}
```

**Example 39**

```java
@Override
public Icon getIcon() {
        int modelCol = header.getTable().convertColumnIndexToModel(curCol);
        TableModel model = header.getTable().getModel();
        if (model instanceof ExtendedJTableSorterModel) {
                ExtendedJTableSorterModel sortModel = (ExtendedJTableSorterModel) model;
                switch (sortModel.getSortingStatus(modelCol)) {
                        case ExtendedJTableSorterModel.ASCENDING:
                                return UIManager.getIcon("Table.ascendingSortIcon");
                        case ExtendedJTableSorterModel.DESCENDING:
                                return UIManager.getIcon("Table.descendingSortIcon");
                        case ExtendedJTableSorterModel.NOT_SORTED:
                        default:
                                return null;
                }
        } else {
                SortKey sortKey = getSortKey(header.getTable().getRowSorter(), modelCol);
                SortOrder sortOrder = sortKey != null ? sortKey.getSortOrder() :
SortOrder.UNSORTED;
                switch (sortOrder) {
                        case ASCENDING:
                                return UIManager.getIcon("Table.ascendingSortIcon");
                        case DESCENDING:
                                return UIManager.getIcon("Table.descendingSortIcon");
                        case UNSORTED:
                        default:
                                return null;
                }
        }
}
```

**Example 40**

```java
public void createDefaultColumnsFromModel() {
    TableModel m = getModel();
    if (m != null) {
        // Remove any current columns
        ProfilerColumnModel cm = _getColumnModel();
        while (cm.getColumnCount() > 0)
            cm.removeColumn(cm.getColumn(0));

        // Create new columns from the data model info
        for (int i = 0; i < m.getColumnCount(); i++)
            addColumn(cm.createTableColumn(i));
    }
}
```