| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultTableModel.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/table/DefaultTableColumnModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/table/JTableHeader.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/table/DefaultTableModel.html)    [**NO FRAMES**](http://docs.google.com/DefaultTableModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#26in1rg) | [METHOD](#3j2qqm3) |

## **javax.swing.table**

Class DefaultTableModel

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.table.AbstractTableModel](http://docs.google.com/javax/swing/table/AbstractTableModel.html)  
 **javax.swing.table.DefaultTableModel**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [TableModel](http://docs.google.com/javax/swing/table/TableModel.html)

public class **DefaultTableModel**extends [AbstractTableModel](http://docs.google.com/javax/swing/table/AbstractTableModel.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html)

This is an implementation of TableModel that uses a Vector of Vectors to store the cell value objects.

**Warning:** DefaultTableModel returns a column class of Object. When DefaultTableModel is used with a TableRowSorter this will result in extensive use of toString, which for non-String data types is expensive. If you use DefaultTableModel with a TableRowSorter you are strongly encouraged to override getColumnClass to return the appropriate type.

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

**See Also:**[TableModel](http://docs.google.com/javax/swing/table/TableModel.html), [getDataVector()](http://docs.google.com/javax/swing/table/DefaultTableModel.html#getDataVector())

| **Field Summary** | |
| --- | --- |
| protected  [Vector](http://docs.google.com/java/util/Vector.html) | [**columnIdentifiers**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#columnIdentifiers)            The Vector of column identifiers. |
| protected  [Vector](http://docs.google.com/java/util/Vector.html) | [**dataVector**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#dataVector)            The Vector of Vectors of Object values. |

| **Fields inherited from class javax.swing.table.**[**AbstractTableModel**](http://docs.google.com/javax/swing/table/AbstractTableModel.html) |
| --- |
| [listenerList](http://docs.google.com/javax/swing/table/AbstractTableModel.html#listenerList) |

| **Constructor Summary** | |
| --- | --- |
| [**DefaultTableModel**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#DefaultTableModel())()            Constructs a default DefaultTableModel which is a table of zero columns and zero rows. |
| [**DefaultTableModel**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#DefaultTableModel(int,%20int))(int rowCount, int columnCount)            Constructs a DefaultTableModel with rowCount and columnCount of null object values. |
| [**DefaultTableModel**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#DefaultTableModel(java.lang.Object%5B%5D%5B%5D,%20java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html)[][] data, [Object](http://docs.google.com/java/lang/Object.html)[] columnNames)            Constructs a DefaultTableModel and initializes the table by passing data and columnNames to the setDataVector method. |
| [**DefaultTableModel**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#DefaultTableModel(java.lang.Object%5B%5D,%20int))([Object](http://docs.google.com/java/lang/Object.html)[] columnNames, int rowCount)            Constructs a DefaultTableModel with as many columns as there are elements in columnNames and rowCount of null object values. |
| [**DefaultTableModel**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#DefaultTableModel(java.util.Vector,%20int))([Vector](http://docs.google.com/java/util/Vector.html) columnNames, int rowCount)            Constructs a DefaultTableModel with as many columns as there are elements in columnNames and rowCount of null object values. |
| [**DefaultTableModel**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#DefaultTableModel(java.util.Vector,%20java.util.Vector))([Vector](http://docs.google.com/java/util/Vector.html) data, [Vector](http://docs.google.com/java/util/Vector.html) columnNames)            Constructs a DefaultTableModel and initializes the table by passing data and columnNames to the setDataVector method. |

| **Method Summary** | |
| --- | --- |
| void | [**addColumn**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#addColumn(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) columnName)            Adds a column to the model. |
| void | [**addColumn**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#addColumn(java.lang.Object,%20java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html) columnName, [Object](http://docs.google.com/java/lang/Object.html)[] columnData)            Adds a column to the model. |
| void | [**addColumn**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#addColumn(java.lang.Object,%20java.util.Vector))([Object](http://docs.google.com/java/lang/Object.html) columnName, [Vector](http://docs.google.com/java/util/Vector.html) columnData)            Adds a column to the model. |
| void | [**addRow**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#addRow(java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html)[] rowData)            Adds a row to the end of the model. |
| void | [**addRow**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#addRow(java.util.Vector))([Vector](http://docs.google.com/java/util/Vector.html) rowData)            Adds a row to the end of the model. |
| protected static [Vector](http://docs.google.com/java/util/Vector.html) | [**convertToVector**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#convertToVector(java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html)[] anArray)            Returns a vector that contains the same objects as the array. |
| protected static [Vector](http://docs.google.com/java/util/Vector.html) | [**convertToVector**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#convertToVector(java.lang.Object%5B%5D%5B%5D))([Object](http://docs.google.com/java/lang/Object.html)[][] anArray)            Returns a vector of vectors that contains the same objects as the array. |
| int | [**getColumnCount**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#getColumnCount())()            Returns the number of columns in this data table. |
| [String](http://docs.google.com/java/lang/String.html) | [**getColumnName**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#getColumnName(int))(int column)            Returns the column name. |
| [Vector](http://docs.google.com/java/util/Vector.html) | [**getDataVector**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#getDataVector())()            Returns the Vector of Vectors that contains the table's data values. |
| int | [**getRowCount**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#getRowCount())()            Returns the number of rows in this data table. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getValueAt**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#getValueAt(int,%20int))(int row, int column)            Returns an attribute value for the cell at row and column. |
| void | [**insertRow**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#insertRow(int,%20java.lang.Object%5B%5D))(int row, [Object](http://docs.google.com/java/lang/Object.html)[] rowData)            Inserts a row at row in the model. |
| void | [**insertRow**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#insertRow(int,%20java.util.Vector))(int row, [Vector](http://docs.google.com/java/util/Vector.html) rowData)            Inserts a row at row in the model. |
| boolean | [**isCellEditable**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#isCellEditable(int,%20int))(int row, int column)            Returns true regardless of parameter values. |
| void | [**moveRow**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#moveRow(int,%20int,%20int))(int start, int end, int to)            Moves one or more rows from the inclusive range start to end to the to position in the model. |
| void | [**newDataAvailable**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#newDataAvailable(javax.swing.event.TableModelEvent))([TableModelEvent](http://docs.google.com/javax/swing/event/TableModelEvent.html) event)            Equivalent to fireTableChanged. |
| void | [**newRowsAdded**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#newRowsAdded(javax.swing.event.TableModelEvent))([TableModelEvent](http://docs.google.com/javax/swing/event/TableModelEvent.html) e)            Ensures that the new rows have the correct number of columns. |
| void | [**removeRow**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#removeRow(int))(int row)            Removes the row at row from the model. |
| void | [**rowsRemoved**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#rowsRemoved(javax.swing.event.TableModelEvent))([TableModelEvent](http://docs.google.com/javax/swing/event/TableModelEvent.html) event)            Equivalent to fireTableChanged. |
| void | [**setColumnCount**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setColumnCount(int))(int columnCount)            Sets the number of columns in the model. |
| void | [**setColumnIdentifiers**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setColumnIdentifiers(java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html)[] newIdentifiers)            Replaces the column identifiers in the model. |
| void | [**setColumnIdentifiers**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setColumnIdentifiers(java.util.Vector))([Vector](http://docs.google.com/java/util/Vector.html) columnIdentifiers)            Replaces the column identifiers in the model. |
| void | [**setDataVector**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setDataVector(java.lang.Object%5B%5D%5B%5D,%20java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html)[][] dataVector, [Object](http://docs.google.com/java/lang/Object.html)[] columnIdentifiers)            Replaces the value in the dataVector instance variable with the values in the array dataVector. |
| void | [**setDataVector**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setDataVector(java.util.Vector,%20java.util.Vector))([Vector](http://docs.google.com/java/util/Vector.html) dataVector, [Vector](http://docs.google.com/java/util/Vector.html) columnIdentifiers)            Replaces the current dataVector instance variable with the new Vector of rows, dataVector. |
| void | [**setNumRows**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setNumRows(int))(int rowCount)            Obsolete as of Java 2 platform v1.3. |
| void | [**setRowCount**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setRowCount(int))(int rowCount)            Sets the number of rows in the model. |
| void | [**setValueAt**](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setValueAt(java.lang.Object,%20int,%20int))([Object](http://docs.google.com/java/lang/Object.html) aValue, int row, int column)            Sets the object value for the cell at column and row. |

| **Methods inherited from class javax.swing.table.**[**AbstractTableModel**](http://docs.google.com/javax/swing/table/AbstractTableModel.html) |
| --- |
| [addTableModelListener](http://docs.google.com/javax/swing/table/AbstractTableModel.html#addTableModelListener(javax.swing.event.TableModelListener)), [findColumn](http://docs.google.com/javax/swing/table/AbstractTableModel.html#findColumn(java.lang.String)), [fireTableCellUpdated](http://docs.google.com/javax/swing/table/AbstractTableModel.html#fireTableCellUpdated(int,%20int)), [fireTableChanged](http://docs.google.com/javax/swing/table/AbstractTableModel.html#fireTableChanged(javax.swing.event.TableModelEvent)), [fireTableDataChanged](http://docs.google.com/javax/swing/table/AbstractTableModel.html#fireTableDataChanged()), [fireTableRowsDeleted](http://docs.google.com/javax/swing/table/AbstractTableModel.html#fireTableRowsDeleted(int,%20int)), [fireTableRowsInserted](http://docs.google.com/javax/swing/table/AbstractTableModel.html#fireTableRowsInserted(int,%20int)), [fireTableRowsUpdated](http://docs.google.com/javax/swing/table/AbstractTableModel.html#fireTableRowsUpdated(int,%20int)), [fireTableStructureChanged](http://docs.google.com/javax/swing/table/AbstractTableModel.html#fireTableStructureChanged()), [getColumnClass](http://docs.google.com/javax/swing/table/AbstractTableModel.html#getColumnClass(int)), [getListeners](http://docs.google.com/javax/swing/table/AbstractTableModel.html#getListeners(java.lang.Class)), [getTableModelListeners](http://docs.google.com/javax/swing/table/AbstractTableModel.html#getTableModelListeners()), [removeTableModelListener](http://docs.google.com/javax/swing/table/AbstractTableModel.html#removeTableModelListener(javax.swing.event.TableModelListener)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### dataVector

protected [Vector](http://docs.google.com/java/util/Vector.html) **dataVector**

The Vector of Vectors of Object values.

### columnIdentifiers

protected [Vector](http://docs.google.com/java/util/Vector.html) **columnIdentifiers**

The Vector of column identifiers.

| **Constructor Detail** |
| --- |

### DefaultTableModel

public **DefaultTableModel**()

Constructs a default DefaultTableModel which is a table of zero columns and zero rows.

### DefaultTableModel

public **DefaultTableModel**(int rowCount,  
 int columnCount)

Constructs a DefaultTableModel with rowCount and columnCount of null object values.

**Parameters:**rowCount - the number of rows the table holdscolumnCount - the number of columns the table holds**See Also:**[setValueAt(java.lang.Object, int, int)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setValueAt(java.lang.Object,%20int,%20int))

### DefaultTableModel

public **DefaultTableModel**([Vector](http://docs.google.com/java/util/Vector.html) columnNames,  
 int rowCount)

Constructs a DefaultTableModel with as many columns as there are elements in columnNames and rowCount of null object values. Each column's name will be taken from the columnNames vector.

**Parameters:**columnNames - vector containing the names of the new columns; if this is null then the model has no columnsrowCount - the number of rows the table holds**See Also:**[setDataVector(java.util.Vector, java.util.Vector)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setDataVector(java.util.Vector,%20java.util.Vector)), [setValueAt(java.lang.Object, int, int)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setValueAt(java.lang.Object,%20int,%20int))

### DefaultTableModel

public **DefaultTableModel**([Object](http://docs.google.com/java/lang/Object.html)[] columnNames,  
 int rowCount)

Constructs a DefaultTableModel with as many columns as there are elements in columnNames and rowCount of null object values. Each column's name will be taken from the columnNames array.

**Parameters:**columnNames - array containing the names of the new columns; if this is null then the model has no columnsrowCount - the number of rows the table holds**See Also:**[setDataVector(java.util.Vector, java.util.Vector)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setDataVector(java.util.Vector,%20java.util.Vector)), [setValueAt(java.lang.Object, int, int)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setValueAt(java.lang.Object,%20int,%20int))

### DefaultTableModel

public **DefaultTableModel**([Vector](http://docs.google.com/java/util/Vector.html) data,  
 [Vector](http://docs.google.com/java/util/Vector.html) columnNames)

Constructs a DefaultTableModel and initializes the table by passing data and columnNames to the setDataVector method.

**Parameters:**data - the data of the table, a Vector of Vectors of Object valuescolumnNames - vector containing the names of the new columns**See Also:**[getDataVector()](http://docs.google.com/javax/swing/table/DefaultTableModel.html#getDataVector()), [setDataVector(java.util.Vector, java.util.Vector)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setDataVector(java.util.Vector,%20java.util.Vector))

### DefaultTableModel

public **DefaultTableModel**([Object](http://docs.google.com/java/lang/Object.html)[][] data,  
 [Object](http://docs.google.com/java/lang/Object.html)[] columnNames)

Constructs a DefaultTableModel and initializes the table by passing data and columnNames to the setDataVector method. The first index in the Object[][] array is the row index and the second is the column index.

**Parameters:**data - the data of the tablecolumnNames - the names of the columns**See Also:**[getDataVector()](http://docs.google.com/javax/swing/table/DefaultTableModel.html#getDataVector()), [setDataVector(java.util.Vector, java.util.Vector)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setDataVector(java.util.Vector,%20java.util.Vector))

| **Method Detail** |
| --- |

### getDataVector

public [Vector](http://docs.google.com/java/util/Vector.html) **getDataVector**()

Returns the Vector of Vectors that contains the table's data values. The vectors contained in the outer vector are each a single row of values. In other words, to get to the cell at row 1, column 5:

((Vector)getDataVector().elementAt(1)).elementAt(5);

**Returns:**the vector of vectors containing the tables data values**See Also:**[newDataAvailable(javax.swing.event.TableModelEvent)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#newDataAvailable(javax.swing.event.TableModelEvent)), [newRowsAdded(javax.swing.event.TableModelEvent)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#newRowsAdded(javax.swing.event.TableModelEvent)), [setDataVector(java.util.Vector, java.util.Vector)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setDataVector(java.util.Vector,%20java.util.Vector))

### setDataVector

public void **setDataVector**([Vector](http://docs.google.com/java/util/Vector.html) dataVector,  
 [Vector](http://docs.google.com/java/util/Vector.html) columnIdentifiers)

Replaces the current dataVector instance variable with the new Vector of rows, dataVector. Each row is represented in dataVector as a Vector of Object values. columnIdentifiers are the names of the new columns. The first name in columnIdentifiers is mapped to column 0 in dataVector. Each row in dataVector is adjusted to match the number of columns in columnIdentifiers either by truncating the Vector if it is too long, or adding null values if it is too short.

Note that passing in a null value for dataVector results in unspecified behavior, an possibly an exception.

**Parameters:**dataVector - the new data vectorcolumnIdentifiers - the names of the columns**See Also:**[getDataVector()](http://docs.google.com/javax/swing/table/DefaultTableModel.html#getDataVector())

### setDataVector

public void **setDataVector**([Object](http://docs.google.com/java/lang/Object.html)[][] dataVector,  
 [Object](http://docs.google.com/java/lang/Object.html)[] columnIdentifiers)

Replaces the value in the dataVector instance variable with the values in the array dataVector. The first index in the Object[][] array is the row index and the second is the column index. columnIdentifiers are the names of the new columns.

**Parameters:**dataVector - the new data vectorcolumnIdentifiers - the names of the columns**See Also:**[setDataVector(Vector, Vector)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setDataVector(java.util.Vector,%20java.util.Vector))

### newDataAvailable

public void **newDataAvailable**([TableModelEvent](http://docs.google.com/javax/swing/event/TableModelEvent.html) event)

Equivalent to fireTableChanged.

**Parameters:**event - the change event

### newRowsAdded

public void **newRowsAdded**([TableModelEvent](http://docs.google.com/javax/swing/event/TableModelEvent.html) e)

Ensures that the new rows have the correct number of columns. This is accomplished by using the setSize method in Vector which truncates vectors which are too long, and appends nulls if they are too short. This method also sends out a tableChanged notification message to all the listeners.

**Parameters:**e - this TableModelEvent describes where the rows were added. If null it assumes all the rows were newly added**See Also:**[getDataVector()](http://docs.google.com/javax/swing/table/DefaultTableModel.html#getDataVector())

### rowsRemoved

public void **rowsRemoved**([TableModelEvent](http://docs.google.com/javax/swing/event/TableModelEvent.html) event)

Equivalent to fireTableChanged.

**Parameters:**event - the change event

### setNumRows

public void **setNumRows**(int rowCount)

Obsolete as of Java 2 platform v1.3. Please use setRowCount instead.

### setRowCount

public void **setRowCount**(int rowCount)

Sets the number of rows in the model. If the new size is greater than the current size, new rows are added to the end of the model If the new size is less than the current size, all rows at index rowCount and greater are discarded.

**Since:** 1.3 **See Also:**[setColumnCount(int)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setColumnCount(int))

### addRow

public void **addRow**([Vector](http://docs.google.com/java/util/Vector.html) rowData)

Adds a row to the end of the model. The new row will contain null values unless rowData is specified. Notification of the row being added will be generated.

**Parameters:**rowData - optional data of the row being added

### addRow

public void **addRow**([Object](http://docs.google.com/java/lang/Object.html)[] rowData)

Adds a row to the end of the model. The new row will contain null values unless rowData is specified. Notification of the row being added will be generated.

**Parameters:**rowData - optional data of the row being added

### insertRow

public void **insertRow**(int row,  
 [Vector](http://docs.google.com/java/util/Vector.html) rowData)

Inserts a row at row in the model. The new row will contain null values unless rowData is specified. Notification of the row being added will be generated.

**Parameters:**row - the row index of the row to be insertedrowData - optional data of the row being added **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the row was invalid

### insertRow

public void **insertRow**(int row,  
 [Object](http://docs.google.com/java/lang/Object.html)[] rowData)

Inserts a row at row in the model. The new row will contain null values unless rowData is specified. Notification of the row being added will be generated.

**Parameters:**row - the row index of the row to be insertedrowData - optional data of the row being added **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the row was invalid

### moveRow

public void **moveRow**(int start,  
 int end,  
 int to)

Moves one or more rows from the inclusive range start to end to the to position in the model. After the move, the row that was at index start will be at index to. This method will send a tableChanged notification message to all the listeners.

Examples of moves:

1. moveRow(1,3,5);  
 a|B|C|D|e|f|g|h|i|j|k - before  
 a|e|f|g|h|B|C|D|i|j|k - after

2. moveRow(6,7,1);  
 a|b|c|d|e|f|G|H|i|j|k - before  
 a|G|H|b|c|d|e|f|i|j|k - after

**Parameters:**start - the starting row index to be movedend - the ending row index to be movedto - the destination of the rows to be moved **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if any of the elements would be moved out of the table's range

### removeRow

public void **removeRow**(int row)

Removes the row at row from the model. Notification of the row being removed will be sent to all the listeners.

**Parameters:**row - the row index of the row to be removed **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the row was invalid

### setColumnIdentifiers

public void **setColumnIdentifiers**([Vector](http://docs.google.com/java/util/Vector.html) columnIdentifiers)

Replaces the column identifiers in the model. If the number of newIdentifiers is greater than the current number of columns, new columns are added to the end of each row in the model. If the number of newIdentifiers is less than the current number of columns, all the extra columns at the end of a row are discarded.

**Parameters:**columnIdentifiers - vector of column identifiers. If null, set the model to zero columns**See Also:**[setNumRows(int)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setNumRows(int))

### setColumnIdentifiers

public void **setColumnIdentifiers**([Object](http://docs.google.com/java/lang/Object.html)[] newIdentifiers)

Replaces the column identifiers in the model. If the number of newIdentifiers is greater than the current number of columns, new columns are added to the end of each row in the model. If the number of newIdentifiers is less than the current number of columns, all the extra columns at the end of a row are discarded.

**Parameters:**newIdentifiers - array of column identifiers. If null, set the model to zero columns**See Also:**[setNumRows(int)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setNumRows(int))

### setColumnCount

public void **setColumnCount**(int columnCount)

Sets the number of columns in the model. If the new size is greater than the current size, new columns are added to the end of the model with null cell values. If the new size is less than the current size, all columns at index columnCount and greater are discarded.

**Parameters:**columnCount - the new number of columns in the model**Since:** 1.3 **See Also:**[setColumnCount(int)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setColumnCount(int))

### addColumn

public void **addColumn**([Object](http://docs.google.com/java/lang/Object.html) columnName)

Adds a column to the model. The new column will have the identifier columnName, which may be null. This method will send a tableChanged notification message to all the listeners. This method is a cover for addColumn(Object, Vector) which uses null as the data vector.

**Parameters:**columnName - the identifier of the column being added

### addColumn

public void **addColumn**([Object](http://docs.google.com/java/lang/Object.html) columnName,  
 [Vector](http://docs.google.com/java/util/Vector.html) columnData)

Adds a column to the model. The new column will have the identifier columnName, which may be null. columnData is the optional vector of data for the column. If it is null the column is filled with null values. Otherwise, the new data will be added to model starting with the first element going to row 0, etc. This method will send a tableChanged notification message to all the listeners.

**Parameters:**columnName - the identifier of the column being addedcolumnData - optional data of the column being added

### addColumn

public void **addColumn**([Object](http://docs.google.com/java/lang/Object.html) columnName,  
 [Object](http://docs.google.com/java/lang/Object.html)[] columnData)

Adds a column to the model. The new column will have the identifier columnName. columnData is the optional array of data for the column. If it is null the column is filled with null values. Otherwise, the new data will be added to model starting with the first element going to row 0, etc. This method will send a tableChanged notification message to all the listeners.

**See Also:**[addColumn(Object, Vector)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#addColumn(java.lang.Object,%20java.util.Vector))

### getRowCount

public int **getRowCount**()

Returns the number of rows in this data table.

**Specified by:**[getRowCount](http://docs.google.com/javax/swing/table/TableModel.html#getRowCount()) in interface [TableModel](http://docs.google.com/javax/swing/table/TableModel.html) **Returns:**the number of rows in the model**See Also:**[TableModel.getColumnCount()](http://docs.google.com/javax/swing/table/TableModel.html#getColumnCount())

### getColumnCount

public int **getColumnCount**()

Returns the number of columns in this data table.

**Specified by:**[getColumnCount](http://docs.google.com/javax/swing/table/TableModel.html#getColumnCount()) in interface [TableModel](http://docs.google.com/javax/swing/table/TableModel.html) **Returns:**the number of columns in the model**See Also:**[TableModel.getRowCount()](http://docs.google.com/javax/swing/table/TableModel.html#getRowCount())

### getColumnName

public [String](http://docs.google.com/java/lang/String.html) **getColumnName**(int column)

Returns the column name.

**Specified by:**[getColumnName](http://docs.google.com/javax/swing/table/TableModel.html#getColumnName(int)) in interface [TableModel](http://docs.google.com/javax/swing/table/TableModel.html)**Overrides:**[getColumnName](http://docs.google.com/javax/swing/table/AbstractTableModel.html#getColumnName(int)) in class [AbstractTableModel](http://docs.google.com/javax/swing/table/AbstractTableModel.html) **Parameters:**column - the column being queried **Returns:**a name for this column using the string value of the appropriate member in columnIdentifiers. If columnIdentifiers does not have an entry for this index, returns the default name provided by the superclass.

### isCellEditable

public boolean **isCellEditable**(int row,  
 int column)

Returns true regardless of parameter values.

**Specified by:**[isCellEditable](http://docs.google.com/javax/swing/table/TableModel.html#isCellEditable(int,%20int)) in interface [TableModel](http://docs.google.com/javax/swing/table/TableModel.html)**Overrides:**[isCellEditable](http://docs.google.com/javax/swing/table/AbstractTableModel.html#isCellEditable(int,%20int)) in class [AbstractTableModel](http://docs.google.com/javax/swing/table/AbstractTableModel.html) **Parameters:**row - the row whose value is to be queriedcolumn - the column whose value is to be queried **Returns:**true**See Also:**[setValueAt(java.lang.Object, int, int)](http://docs.google.com/javax/swing/table/DefaultTableModel.html#setValueAt(java.lang.Object,%20int,%20int))

### getValueAt

public [Object](http://docs.google.com/java/lang/Object.html) **getValueAt**(int row,  
 int column)

Returns an attribute value for the cell at row and column.

**Specified by:**[getValueAt](http://docs.google.com/javax/swing/table/TableModel.html#getValueAt(int,%20int)) in interface [TableModel](http://docs.google.com/javax/swing/table/TableModel.html) **Parameters:**row - the row whose value is to be queriedcolumn - the column whose value is to be queried **Returns:**the value Object at the specified cell **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if an invalid row or column was given

### setValueAt

public void **setValueAt**([Object](http://docs.google.com/java/lang/Object.html) aValue,  
 int row,  
 int column)

Sets the object value for the cell at column and row. aValue is the new value. This method will generate a tableChanged notification.

**Specified by:**[setValueAt](http://docs.google.com/javax/swing/table/TableModel.html#setValueAt(java.lang.Object,%20int,%20int)) in interface [TableModel](http://docs.google.com/javax/swing/table/TableModel.html)**Overrides:**[setValueAt](http://docs.google.com/javax/swing/table/AbstractTableModel.html#setValueAt(java.lang.Object,%20int,%20int)) in class [AbstractTableModel](http://docs.google.com/javax/swing/table/AbstractTableModel.html) **Parameters:**aValue - the new value; this can be nullrow - the row whose value is to be changedcolumn - the column whose value is to be changed **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if an invalid row or column was given**See Also:**[TableModel.getValueAt(int, int)](http://docs.google.com/javax/swing/table/TableModel.html#getValueAt(int,%20int)), [TableModel.isCellEditable(int, int)](http://docs.google.com/javax/swing/table/TableModel.html#isCellEditable(int,%20int))

### convertToVector

protected static [Vector](http://docs.google.com/java/util/Vector.html) **convertToVector**([Object](http://docs.google.com/java/lang/Object.html)[] anArray)

Returns a vector that contains the same objects as the array.

**Parameters:**anArray - the array to be converted **Returns:**the new vector; if anArray is null, returns null

### convertToVector

protected static [Vector](http://docs.google.com/java/util/Vector.html) **convertToVector**([Object](http://docs.google.com/java/lang/Object.html)[][] anArray)

Returns a vector of vectors that contains the same objects as the array.

**Parameters:**anArray - the double array to be converted **Returns:**the new vector of vectors; if anArray is null, returns null

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultTableModel.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/table/DefaultTableColumnModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/table/JTableHeader.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/table/DefaultTableModel.html)    [**NO FRAMES**](http://docs.google.com/DefaultTableModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#26in1rg) | [METHOD](#3j2qqm3) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).