目录

[阅读有关说明4](#_Toc261425203)

[常见变量、类及方法： 11](#_Toc261425204)

[com.mxgraph.analysis包： 12](#_Toc261425205)

[此包为图的分析提供了各种算法，例如最短路径和最小生成树 12](#_Toc261425206)

[com.mxgraph.canvas包： 12](#_Toc261425207)

[此包包含了使用不同技术绘制graph的各种工具，例如：graphics2D、html、SVG、VML等 12](#_Toc261425208)

[com.mxgraph.io包 12](#_Toc261425209)

[此包包含了输入输出所用到的所有类 12](#_Toc261425210)

[com.mxgraph.layout 12](#_Toc261425211)

[此包包含各种graph的布局 12](#_Toc261425212)

[com.mxgraph.layout.hierarchical 13](#_Toc261425213)

[com.mxgraph.layout.hierarchical.model 13](#_Toc261425214)

[com.mxgraph.layout.hierarchical.stage 13](#_Toc261425215)

[com.mxgraph.layout.orthogonal 13](#_Toc261425216)

[com.mxgraph.layout.orthogonal.model 13](#_Toc261425217)

[\*com.mxgraph.model 包： 13](#_Toc261425218)

[此包包含定义graph model的各个类 13](#_Toc261425219)

[\*com.mxgraph.model 中mxCell类: 13](#_Toc261425220)

[\*com.mxgraph.model 中mxcellpath类： 13](#_Toc261425221)

[\*com.mxgraph.model 中mxGeometry类: 14](#_Toc261425222)

[\*com.mxgraph.model 中mxGraphModel类: 14](#_Toc261425223)

[\*com.mxgraph.model 中mxIgraphmodel接口: 14](#_Toc261425224)

[\*com.mxgraph.model包中mxIGraphModel.mxAtomicGraphModelChange类： 16](#_Toc261425225)

[\*com.mxgraph.model 包中mxIcell接口： 16](#_Toc261425226)

[\*com.mxgraph.model 包中mxIcell接口： 16](#_Toc261425227)

[com.mxgraph.reader包 17](#_Toc261425228)

[此包包含的类有如下功能：使用SAX将一个编码过的mxGraphView转化为一个image，并且此过程不用创造一个graph model 17](#_Toc261425229)

[com.mxgraph.sharing 17](#_Toc261425230)

[这个软件包包含了由多个客户端并发图编辑所需的所有类。 17](#_Toc261425231)

[com.mxgraph.swing包： 17](#_Toc261425232)

[包含了JFC或Swing的主要成分，也就是graph部分和外形部分 17](#_Toc261425233)

[com.mxgraph.swing.handler 包： 17](#_Toc261425234)

[包含所有用于处理JFC/Swing鼠标响应事件的类 17](#_Toc261425235)

[Classes 17](#_Toc261425236)

[com.mxgraph.swing.util 包： 18](#_Toc261425237)

[包含需要JFC/Swing的所有有用的类，也就是：对鼠标处理事件、拖拽和删除事件以及图标等。 18](#_Toc261425238)

[com.mxgraph.swing.view 包： 18](#_Toc261425239)

[这个包包含了交互所需要的所有类 18](#_Toc261425240)

[com.mxgraph.util 包： 19](#_Toc261425241)

[com.mxgraph.view包： 19](#_Toc261425242)

[实现graph的component。 20](#_Toc261425243)

[Mxcellstate类： 20](#_Toc261425244)

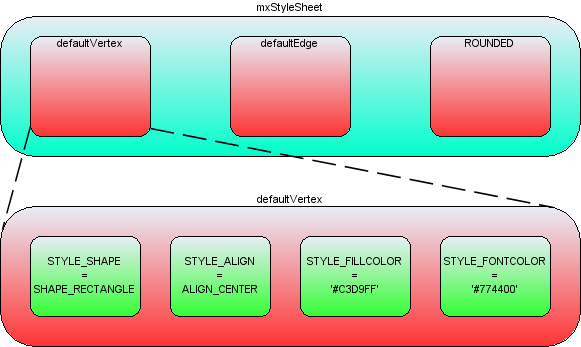
[有关label的问题： 21](#_Toc261425245)

jgraphx  
  .......\build.xml  
  .......\docs  
  .......\....\allclasses-frame.html  
  .......\....\allclasses-noframe.html  
  .......\....\com  
  .......\....\...\mxgraph  
  .......\....\...\.......\algebra  
  .......\....\...\.......\.......\class-use  
  .......\....\...\.......\.......\.........\mxConstantCostFunction.html  
  .......\....\...\.......\.......\.........\mxDistanceCostFunction.html  
  .......\....\...\.......\.......\.........\mxFibonacciHeap.html  
  .......\....\...\.......\.......\.........\mxFibonacciHeap.Node.html  
  .......\....\...\.......\.......\.........\mxGraphAlgebra.html  
  .......\....\...\.......\.......\.........\mxICostFunction.html  
  .......\....\...\.......\.......\.........\mxUnionFind.html  
  .......\....\...\.......\.......\.........\mxUnionFind.Node.html  
  .......\....\...\.......\.......\mxConstantCostFunction.html  
  .......\....\...\.......\.......\mxDistanceCostFunction.html  
  .......\....\...\.......\.......\mxFibonacciHeap.html  
  .......\....\...\.......\.......\mxFibonacciHeap.Node.html  
  .......\....\...\.......\.......\mxGraphAlgebra.html  
  .......\....\...\.......\.......\mxICostFunction.html  
  .......\....\...\.......\.......\mxUnionFind.html  
  .......\....\...\.......\.......\mxUnionFind.Node.html  
  .......\....\...\.......\.......\package-frame.html  
  .......\....\...\.......\.......\package-summary.html  
  .......\....\...\.......\.......\package-tree.html  
  .......\....\...\.......\.......\package-use.html  
  .......\....\...\.......\canvas  
  .......\....\...\.......\......\class-use  
  .......\....\...\.......\......\.........\mxBasicCanvas.html  
  .......\....\...\.......\......\.........\mxGraphics2DCanvas.html  
  .......\....\...\.......\......\.........\mxHtmlCanvas.html  
  .......\....\...\.......\......\.........\mxICanvas.html  
  .......\....\...\.......\......\.........\mxImageCanvas.html  
  .......\....\...\.......\......\.........\mxSvgCanvas.html  
  .......\....\...\.......\......\.........\mxVmlCanvas.html  
  .......\....\...\.......\......\mxBasicCanvas.html  
  .......\....\...\.......\......\mxGraphics2DCanvas.html  
  .......\....\...\.......\......\mxHtmlCanvas.html  
  .......\....\...\.......\......\mxICanvas.html  
  .......\....\...\.......\......\mxImageCanvas.html  
  .......\....\...\.......\......\mxSvgCanvas.html  
  .......\....\...\.......\......\mxVmlCanvas.html  
  .......\....\...\.......\......\package-frame.html  
  .......\....\...\.......\......\package-summary.html  
  .......\....\...\.......\......\package-tree.html  
  .......\....\...\.......\......\package-use.html  
  .......\....\...\.......\io  
  .......\....\...\.......\..\class-use  
  .......\....\...\.......\..\.........\mxCellCodec.html  
  .......\....\...\.......\..\.........\mxCodec.html  
  .......\....\...\.......\..\.........\mxCodecRegistry.html  
  .......\....\...\.......\..\.........\mxModelCodec.html  
  .......\....\...\.......\..\.........\mxObjectCodec.html  
  .......\....\...\.......\..\.........\mxStylesheetCodec.html  
  .......\....\...\.......\..\mxCellCodec.html  
  .......\....\...\.......\..\mxCodec.html  
  .......\....\...\.......\..\mxCodecRegistry.html  
  .......\....\...\.......\..\mxModelCodec.html  
  .......\....\...\.......\..\mxObjectCodec.html  
  .......\....\...\.......\..\mxStylesheetCodec.html  
  .......\....\...\.......\..\package-frame.html  
  .......\....\...\.......\..\package-summary.html  
  .......\....\...\.......\..\package-tree.html  
  .......\....\...\.......\..\package-use.html  
  .......\....\...\.......\layout  
  .......\....\...\.......\......\class-use  
  .......\....\...\.......\......\.........\mxCircleLayout.html  
  .......\....\...\.......\......\.........\mxGraphLayout.html  
  .......\....\...\.......\......\.........\mxIGraphLayout.html  
  .......\....\...\.......\......\.........\mxPartitionLayout.html  
  .......\....\...\.......\......\.........\mxStackLayout.html  
  .......\....\...\.......\......\mxCircleLayout.html  
  .......\....\...\.......\......\mxGraphLayout.html  
  .......\....\...\.......\......\mxIGraphLayout.html  
  .......\....\...\.......\......\mxPartitionLayout.html  
  .......\....\...\.......\......\mxStackLayout.html  
  .......\....\...\.......\......\package-frame.html  
  .......\....\...\.......\......\package-summary.html  
  .......\....\...\.......\......\package-tree.html  
  .......\....\...\.......\......\package-use.html  
  .......\....\...\.......\model  
  .......\....\...\.......\.....\class-use  
  .......\....\...\.......\.....\.........\mxCell.html  
  .......\....\...\.......\.....\.........\mxCellPath.html  
  .......\....\...\.......\.....\.........\mxGeometry.html  
  .......\....\...\.......\.....\.........\mxGraphModel.Filter.html  
  .......\....\...\.......\.....\.........\mxGraphModel.html  
  .......\....\...\.......\.....\.........\mxGraphModel.mxChildChange.html  
  .......\....\...\.......\.....\.........\mxGraphModel.mxCollapseChange.html  
  .......\....\...\.......\.....\.........\mxGraphModel.mxGeometryChange.html  
  .......\....\...\.......\.....\.........\mxGraphModel.mxRootChange.html  
  .......\....\...\.......\.....\.........\mxGraphModel.mxStyleChange.html  
  .......\....\...\.......\.....\.........\mxGraphModel.mxTerminalChange.html  
  .......\....\...\.......\.....\.........\mxGraphModel.mxValueChange.html  
  .......\....\...\.......\.....\.........\mxGraphModel.mxVisibleChange.html  
  .......\....\...\.......\.....\.........\mxICell.html

阅读有关说明  
  .......\....\...\.......\.....\.........\mxIGraphModel.html虽然许多主要API是通过mxGraph类调用，记住mxGraphModel是存储您的图形数据结构的根本对象。  
mxCell复制了model的许多方法，使用它们的主要不同在于：利用model的方法创建适当的事件通知和撤消；利用cell使其变化但不记录变化。也就是说你要用model来了编辑API，除非你遇到了这个机制的具体问题。

当你创建一个新的cell时候，有三个部分需要来构造：value、geometry和style；

mxStyleSheet：类似于css样式表



1. A new style called 'ROUNDED' has been created, to apply this to a vertex:

Object v1 = graph.insertVertex(parent, null, "Hello", 20, 20, 80, 30, "ROUNDED");

1. To create a new vertex with the ROUNDED style, overriding the stroke and fill colors:

Object v1 = graph.insertVertex(parent, null, "Hello", 20, 20, 80, 30, "ROUNDED;strokeColor=red;fillColor=green");

1. To create a new vertex with no global style, but with local stroke and fill colors:

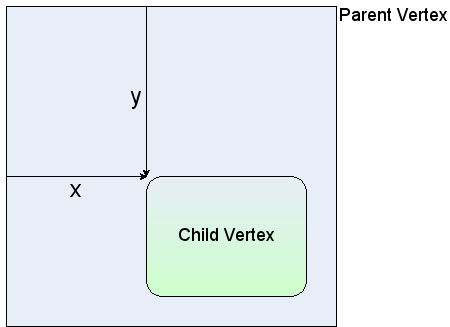
Object v1 = graph.insertVertex(parent, null, "Hello", 20, 20, 80, 30, ";strokeColor=red;fillColor=green");

1. To create a vertex that uses the defaultVertex style, but a local value of the fill color:

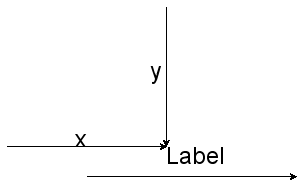
Object v1 = graph.insertVertex(parent, null, "Hello", 20, 20, 80, 30, "defaultVertex;fillColor=blue");

* **mxGraph.setCellStyle(style, cells)** – Sets the style for the array of cells, encapsulated in a begin/end update.
* **mxGraph.getCellStyle(cell)** – Returns the style for the specified cell, merging the styles from any local style and the default style for that cell type. 、

一、坐标在无关位置中：

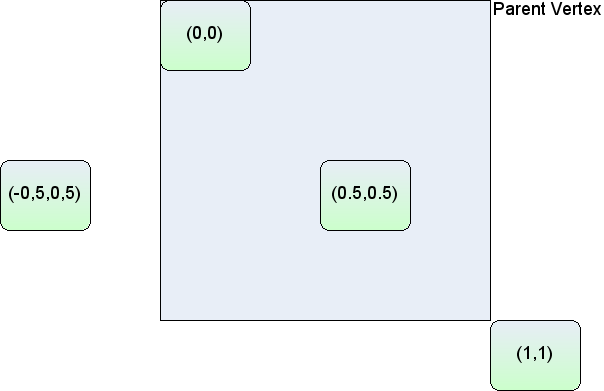


如上是节点中x与y的意思：child相对于parent的x和y

  
Non-relative edge label positioning

如上是边中坐标的意思：在其label的左上角

二、坐标在相关位置中

  
Relative vertex positions

对于相对模式的顶点，第（x，y）是沿着父辈细胞的（宽度的比例，高）在细胞的来源所在。 （0,0）是与父同根同源，（1,1）置于父右下角的起源。同样的相对定位扩展低于0及以上两个方面1。这个定位是非常有用的儿童细胞保持相对固定的整体父单元尺寸。

最后，在相对模式标签palced优势的基础上，从边缘的中心位置。的x坐标是从边缘源端的相对距离为-1，到边的目标年底，1。在y协调是偏离正交边缘的像素。下图显示的x值，y对于不同模式相对优势的标签。请注意，对于一个直边，计算简单。对于那些有多个控制点的边缘，边缘要追溯到沿段（1段是六月底的点和线/或控制点）找到正确沿着边缘的距离。在y值是正交偏移的部分。

交换相对定位上是一个应用程序中常见的偏好边缘标签。导航到mxGraph.insertEdge（）在mxGraph方法，您会看到此呼吁createEdge（）。在createEdge（）的几何设置为每条边的相对创建使用这个原型。这部分是用于辅助方法mxGraph数额，他们可以轻松实现的默认的行为发生变化。你应该尝试使用尽可能地提供给你的应用有利于mxGraph类API。

在mxgeometry中的偏移量是相对于cell中label而言的，至于edge label的情况，偏移量总是设在label计算之后。

**Core API methods:**

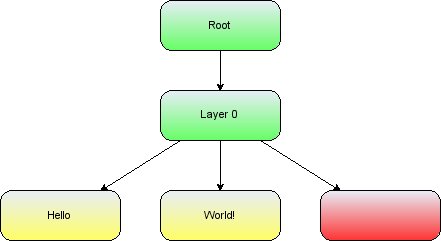
* **mxGraph.resizeCell(cell, bounds)** – Resizes the specified cell to the specified bounds, within a begin/end update call.
* **mxGraph.resizeCells(cells, bounds)** – Resizes each of the cells in the cells array to the corresponding entry in the bounds array, within a begin/end update call.

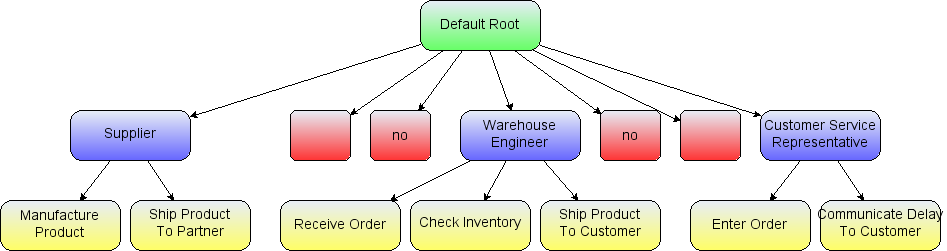
\*用户对象是给JGraphx图表一个上下文，它用一个可见的cell存储业务逻辑；在HelloWorld例子中用户对象只是一个string——字符串，这个例子中它只是代表cell将要展示的label；而在一些更复杂的应用中，这些用户对象将是真正的对象，

这些属性显示的几何形状，标签，身份证等，但一个对话框，可以很容易地显示细胞的用户对象。可能有一些在工作流引擎进程的职权如何存货的实际检查。这可能是一个应用程序在服务器和客户端具体机制，指定某种识别的远程方法调用。另一个值可能是对象的类型程序产生的，也许一个布尔值或整数表明股市在这种情况下的水平。由于返回类型，可以强制执行图的限制，并提供可视化的警报，如果说，即将离任的边缘检查的决定不符合的顶点的返回类型。

即将离任的边缘用户对象可能包含一个标签和一个布尔状态。同样，JGraphX的编辑提供的手段来改变布尔值。在执行的过程中，它可能会按照边对应于布尔值的决策节点返回。

请记住，上面的例子很具体的领域，它的存在说明了用户对象映射到应用程序的业务逻辑。它可视化创造如何JGraphX我们称之为背景图。当前的情况是由顶点之间的业务逻辑的连接形成存储在用户对象。一个典型的应用程序收到的视觉和业务逻辑从断绝，可允许两个编辑，然后传送两回的持久性和/或执行服务器。

  
The group structure of the helloworld example

  
The logical group structure of the workflow example

\*就像我们描述的那样，mxgraph使我们用该library的主要API，对于cell也是这样

\*在mxcell类中存在两个布尔类型的flag：vertex和edge；当一个cell背创建之后一些辅助方法会将它们之中的一个设置为true；model使用mxIGraphModel中的isvertex（）和isedge（）方法来决定一个cell的类型；

\*在mxgraph类中向group structure插入cells可以通过使用insertvertex（）和insertedge（）中parents的参数设置得以实现

\*而我们通过mxGraph.groupCells() 和mxGraph.ungroupCells()方法来改变group structure

**Core API methods:**

* **mxGraph.groupCells(group, border, cells)** – Adds the specified cells to the specified group, within a begin/end update
* **mxGraph.ungroupCells(cells)** – Removes the specified cells from their parent and adds them to their parent's parent. Any group empty after the operation are deleted. The operation occurs within a begin/end update.

\*Folding（折叠）

我们使用折叠去扩展（expanding）和缩小（collapsing），我们认为一个cell将通过设置其子节点为不可见的方式来折叠自己

**Core API method:**

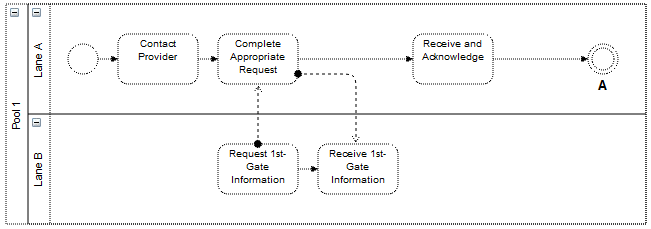
* **mxGraph.foldCells(collapse, recurse, cells)** – States the collapsed state of the specificed cells, within a begin/end update.
* **Folding related methods:**
* **mxGraph.isCellFoldable(cell, collapse)** – By default true for cells with children.
* **mxGraph.isCellCollapsed(cell)** – Returns the folded state of the cell

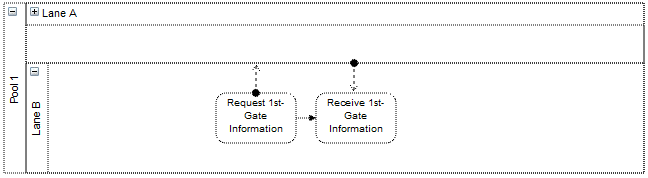
\*当一个group cell被collapse之后，会发生三件事儿：

一、它的子节点会变为不可见

* 二、这个group cell的group bounds被使用，The group bounds of the group cell is used. Within mxGeometry there is a alternativeBounds field and in groups cells, by default store a separate bounds for their collapsed and expanded states. The switch between these instances is invoked by mxGraph.swapBounds() and this is handled for you within a foldCells() call. This allows collapsed groups to be resized whilst when expanded again the size looks correct using the pre-collapsed size.

三、边的提升。边提升的意思是将与collapsed后的 group相连的子节点所使用的边与collapsed后的parents相连以使其显现。

  
*Expanded swimlane*

  
*Collapsed Swimlane*

注意：使用mxGraph.foldCells()函数，你可以得到与使用群组左上角“+”“-”符号相同的效果。

**2.2.5.2 Sub-Graphs, Drill-Down / Step-Up**

\*有时候，作为一种expand或collapse的一种选择，或者用其来做图形整合，你的图形有可能由多个graphs组成，并嵌套成一个层次结构。

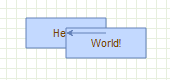
\*鼠标右键中的entergroup将会调用mxGraph.enterGroup(cell)，而该方法是sub-graphs（子图）的核心API方法之一。

**Core API methods:**

* **mxGraph.enterGroup(cell)** – Makes the specified cell the new root of the display area.
* **mxGraph.exitGroup()** - Makes the parent of the current root cell, if any, the new root cell.
* **mxGraph.home()** - Exits all groups, making the default parent the root cell.

2.2.5.3 Layering and Filtering（层次和过滤）

不同的节点有不同的层次：

  
Overlapped vertices

要改变层次我们用方法：mxGraph.orderCells（）

**Core API method:**

* **mxGraph.orderCells(back, cells)** – Moves the array of cells to the front or back of their siblings, depending on the flag, within a begin/end update.（将某cell的层次顺序置于其同胞（同parents）cell之前或之后）

# 常见变量、类及方法：

DefaultParent：指定用来插入cell的初始parents，通常在getdefaultParent（）中使用；在此default是空的意思

MxGraph类：主要类，用来插入节点、边等；如addcells，removecells，insertvertex，getcellbounds，getedge，findtreeroots等

Mximodel类：处理mxgraph所需要的寻找节点、边，复制节点；得到父节点、子节点；得到边，看是否是边、节点；等操作

MxGraphcomponent类：这个类是一个连接着内部mxgraphcontrol的scrollpane（滑动面板），其中的control包含着实际的graph展示，包含方法：getgraph，setgraph，getbackgroundimage，zoomin，zoomout（对面板的放大缩小）scrollCellToVisible（滑动graph使得cell可见），getcellat（在具体位置取得cell）等。

Com.mxgraph.canvas包：包括了用不同技术来绘制graph的各种工具，如graph2D、html等

Mxicanvas接口：定义了为绘制一个graph的vertics和edges而对canvas的需求

Mxbasiccanvas：

Mxgraphic2Dcanvas：canvas的一个用来绘制graphic2D图形的工具（其中包含矩形、圆柱体、云形等各种形状的定义及绘制！！）

#### com.mxgraph.layout.hierarchical.model

注：几个layout包中包含的是各种自动布局类，其中\*. hierarchical.model类中定义了一个接口：[mxGraphHierarchyModel.CellVisitor](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\layout\hierarchical\model\mxGraphHierarchyModel.CellVisitor.html)，该接口定义了在深度优先遍历（dfs）或其他遍历方法中图形信息的操作。

# [**com.mxgraph.analysis**](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\analysis\package-summary.html)包：

## **此包为**图的分析提供了各种算法，例如最短路径和最小生成树

# [com.mxgraph.canvas](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\canvas\package-summary.html)包：

## 此包包含了使用不同技术绘制graph的各种工具，例如：graphics2D、html、SVG、VML等

# [**com.mxgraph.io**](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\io\package-summary.html)包

## 此包包含了输入输出所用到的所有类

[com.mxgraph.layout](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\layout\package-summary.html)包

## 此包包含各种graph的布局

|  |  |
| --- | --- |
| [com.mxgraph.layout.hierarchical](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\layout\hierarchical\package-summary.html) |  |
| [com.mxgraph.layout.hierarchical.model](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\layout\hierarchical\model\package-summary.html) |  |
| [com.mxgraph.layout.hierarchical.stage](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\layout\hierarchical\stage\package-summary.html) |  |
| [com.mxgraph.layout.orthogonal](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\layout\orthogonal\package-summary.html) |  |
| [com.mxgraph.layout.orthogonal.model](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\layout\orthogonal\model\package-summary.html) |  |

# **\***[**com.mxgraph.model**](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\package-summary.html) **包：**

## 此包包含定义graph model的各个类

mxGraph model 是描述graph 的结构的核心模板，graph结构的修改、增加、删除通过graph model API发生，该model规定了一些方法去决定graph的结构，同时也提供了一些方法去设置可视化状态，例如visibility, grouping 和 style。

然而，尽管对model的处理是存储在model类包中的，但JGraphx是这样设置的：主要的公共API是通过MxGraph类实现的。因为“向graph中添加cell”的说法比“向graph的model中添加cell”更自然、直观。使其更直观的是：在model和cells中的方法在graph中也有一份，而graph类中的方法被认为是主要的public API。所以要记住：尽管许多主要的API是通过mxGraph类实现的，但mxGraphModel是存储你的graph数据结构的根本对象。

\*[com.mxgraph.model](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\package-summary.html) 中mxCell类:cells是graph model的元素。它们描述了一个graph中边、定点和群组的各种状态。边上的label的定义，其位置如何确定。

\*[com.mxgraph.model](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\package-summary.html) 中mxcellpath类：对临时的cell的ID实现了一种机制。其中的create()方法创建了所给cell的cell path，即其在通往root的路径中所有cell的index的串联，如"0.0.0.1"

\*[com.mxgraph.model](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\package-summary.html) 中mxGeometry类:描述一个cell的geometry（几何形状），对节点来说，geometry包括x和y坐标还有width和height。对边来说，geometry或者定义边的起、终点，或者定义其终点位置。

\*[com.mxgraph.model](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\package-summary.html) 中mxGraphModel类:继承自mxEventSource以实现一个graph model。Graph model 就像一个围绕着cell的包装，用来存储实际的graph的数据结构。Model就像一个处理事务的包装，对所有事件进行event通知。然而，cell拥有对数据结构更新的原始操作。层（layers）：在model中的cell层必须有一个top-level root cell，它包含了layers（典型的是一个原始的layer），它们反过来又包含layers的top-level root cell，意思是说，每个cell必须在layer中包含着。如果不需要一个新的layer，那么，所有新的cell必须添加到原始（default）layer中去。对于隐藏和展示cell组（group）来说，layer是有用的。该类可处理cell与其孩子与父亲的关系。

\*[com.mxgraph.model](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\package-summary.html) 中mxIgraphmodel接口:

可用于找出children cell的数量、取得child cell等：getChildCount（）、getEdgeAt()、getChildAt()等（mxGraphModel类实现了此接口！！！）

mxGraph model 是描述graph 的结构的核心模板，graph结构的修改、增加、删除通过graph model API发生，该model规定了一些方法去决定graph的结构，同时也提供了一些方法去设置可视化状态，例如visibility, grouping 和 style。

然而，尽管对model的处理是存储在model类包中的，但JGraphx是这样设置的：主要的公共API是通过MxGraph类实现的。因为“向graph中添加cell”的说法比“向graph的model中添加cell”更自然、直观。使其更直观的是：在model和cells中的方法在graph中也有一份，而graph类中的方法被认为是主要的public API。所以要记住：尽管许多主要的API是通过mxGraph类实现的，但mxGraphModel是存储你的graph数据结构的根本对象！！！。

Jgraph建立了一种机制去使得model改变

// Adds cells to the model in a single step

graph.getModel().beginUpdate();

try

{

Object v1 = graph.addVertex(parent, null, "Hello,", 20, 20, 80, 30);

Object v2 = graph.addVertex(parent, null, "World!", 200, 150, 80, 30);

Object e1 = graph.addEdge(parent, null, "", v1, v2);

}

finally

{

// Updates the display

graph.getModel().endUpdate();

}

在上面，每次改变model，你都要先调用beginUpdate()，改变model后再调用endUpdate()以定下改变。

相关主要方法：

**Key API Methods:**

* **mxGraphModel.beginUpdate()** - starts a new transaction or a sub-transaction.
* **mxGraphModel.endUpdate()** - completes a transaction or a sub-transaction.
* **mxGraph.addVertex()** - Adds a new vertex to the specified parent cell.
* **mxGraph.addEdge()** - Adds a new edge to the specified parent cell.

需要指出的是，其实model会随着更新的范围马上发生改变，而没有必要调用beginUpdate()和endUpdate()，但为了避免小错误和个人习惯，最好用之。

注：从beginUpdate()到endUpdate()是update范围。

“It is important to distinguish between functionality that acts on the graph model as part of a compound change and functionality that reacts to atomic graph change events. In the first case, such as for automatic layouting, the functionality takes the model as-is and acts upon it. This method should only be used for parts of compound model changes. All other parts of the application should only react to model change events.“

下面的方法应该在update的范围中被直接或间接调用：

* add(parent, child, index)
* remove(cell)
* setCollapsed(cell, collapsed)
* setGeometry(cell, geometry)
* setRoot(root)
* setStyle(cell, style)
* setTerminal(cell, terminal, isSource)
* setTerminals(edge,source,target)
* setValue(cell, value)
* setVisible(cell, visible)

需要注意的是它们不是核心API方法，但它们也在mxgraph类中。

一些人对可视化信息被model存储感到疑惑，这些属性包括cell的positioning（位置）、visibility和collapsed。Model存储这些属性的初始状态，并提供一个共同place在per\_cell的基础上来设置它们，然而，views能够在per\_view的基础上覆盖它们。Model只不过是这些属性能够被设置成global basis的结构（体系）中的第一个common place。

你可以通过add（）方法在model中添加一个cell，但你要知道，mxGraph.insertVertex()和mxGraph.insertEdge()是添加cell的核心方法。model中的方法要求当添加一个cell时这个cell必须是已经被创建了的，尽管mxGraph.insertVertex()可以为你创建cell

**Core API methods:**

* **mxGraph.insertVertex(parent, id, value, x, y, width, height, style)** – creates and inserts a new vertex into the model, within a begin/end update call.
* **mxGraph.insertEdge(parent, id, value, source, target, style) –** creates and inserts a new edge into the model, within a begin/end update call.

\*[com.mxgraph.model](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\package-summary.html)包中[mxIGraphModel.mxAtomicGraphModelChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\mxIGraphModel.mxAtomicGraphModelChange.html)类：定义一些graph model的自动变化。以下为其子类：

[mxGraphModel.mxChildChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\mxGraphModel.mxChildChange.html)   
[mxGraphModel.mxCollapseChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\mxGraphModel.mxCollapseChange.html)   
[mxGraphModel.mxGeometryChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\mxGraphModel.mxGeometryChange.html)   
[mxGraphModel.mxRootChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\mxGraphModel.mxRootChange.html)   
[mxGraphModel.mxStyleChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\mxGraphModel.mxStyleChange.html)   
[mxGraphModel.mxTerminalChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\mxGraphModel.mxTerminalChange.html)   
[mxGraphModel.mxValueChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\mxGraphModel.mxValueChange.html)   
[mxGraphModel.mxVisibleChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\mxGraphModel.mxVisibleChange.html)

\*[com.mxgraph.model](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\package-summary.html) 包中mxIcell接口：定义一个能在mxgraph model中使用的cell所需要的各种条件、属性。

\*[com.mxgraph.model](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\model\package-summary.html) 包中mxIcell接口：定义一个能在mxGraph中使用的graph model所需要的各种条件、属性。

# [com.mxgraph.reader](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\reader\package-summary.html)包

## 此包包含的类有如下功能：使用SAX将一个编码过的mxGraphView转化为一个image，并且此过程不用创造一个graph model

# [com.mxgraph.sharing](file:///C:\Documents%20and%20Settings\Administrator\桌面\jgraphx\docs\api\com\mxgraph\sharing\package-summary.html)包

## 这个软件包包含了由多个客户端并发图编辑所需的所有类。

# [com.mxgraph.swing](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\package-frame.html)包：

## 包含了JFC或Swing的主要成分，也就是graph部分和外形部分

\*[com.mxgraph.swing](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\package-frame.html) 包中的类：    
[mxGraphComponent](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\mxGraphComponent.html) ：设置背景、对cell的编辑、添加图标（用来显示的ImageIcon）、背景样式、背景颜色等、得到cell、得到graph。

其中importCells（）方法可以在当一个cell是通过数据传输来插入graph的时候的一种快捷方法！！！  
[mxGraphComponent.mxMouseRedirector](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\mxGraphComponent.mxMouseRedirector.html) ：鼠标响应事件！！  
[mxGraphOutline](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\mxGraphOutline.html)：一个具体graph component的轮廓；

# com.mxgraph.swing.handler 包：

## 包含所有用于处理JFC/Swing鼠标响应事件的类

Classes    
[mxCellHandler](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxCellHandler.html)   
[mxCellMarker](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxCellMarker.html)   
[mxCellTracker](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxCellTracker.html)   
[mxConnectionHandler](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxConnectionHandler.html)   
[mxEdgeHandler](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxEdgeHandler.html)   
[mxElbowEdgeHandler](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxElbowEdgeHandler.html)   
[mxGraphHandler](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxGraphHandler.html)   
[mxGraphTransferHandler](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxGraphTransferHandler.html)   
[mxKeyboardHandler](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxKeyboardHandler.html)   
[mxPanningHandler](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxPanningHandler.html)   
[mxRubberband](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxRubberband.html)   
[mxSubHandler](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxSubHandler.html)   
[mxVertexHandler](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\handler\mxVertexHandler.html)

# com.mxgraph.swing.util 包：

## 包含需要JFC/Swing的所有有用的类，也就是：对鼠标处理事件、拖拽和删除事件以及图标等。

|  |
| --- |
| Interfaces   [*mxICellOverlay*](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxICellOverlay.html) |

|  |  |  |
| --- | --- | --- |
| Classes   [mxCellOverlay](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxCellOverlay.html) ：cell的图标：矩形、圆形、书形等 [mxGraphActions](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.html) ：删除、组群、编辑cell等如下：  [mxGraphActions.DeleteAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.DeleteAction.html)  [mxGraphActions.DrillAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.DrillAction.html)  [mxGraphActions.EditAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.EditAction.html)  [mxGraphActions.FoldAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.FoldAction.html)  [mxGraphActions.GroupAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.GroupAction.html)  [mxGraphActions.LayerAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.LayerAction.html)  [mxGraphActions.RemoveFromParentAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.RemoveFromParentAction.html)  [mxGraphActions.SelectAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.SelectAction.html)  [mxGraphActions.UngroupAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.UngroupAction.html)  [mxGraphActions.UpdateGroupBoundsAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.UpdateGroupBoundsAction.html)  [mxGraphActions.ZoomAction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphActions.ZoomAction.html)  [mxGraphTransferable](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxGraphTransferable.html) ： [mxMouseControl](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\util\mxMouseControl.html)：鼠标响应事件的抽象类 [com.mxgraph.swing.view](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\view\package-summary.html) 包：这个包包含了交互所需要的所有类  |  | | --- | | Interfaces   [*mxICellEditor*](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\view\mxICellEditor.html)： |  |  | | --- | | Classes   [mxCellEditor](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\view\mxCellEditor.html) ：用来执行对cell的就地编辑（添加文本！！！）。  如：com.mxgraph.swing.view.[mxCellEditor](eclipse-javadoc:%E2%98%82=MyCD/src%3Ccom.mxgraph.swing.view%7BmxCellEditor.java%E2%98%83mxCellEditor).mxcelleditor（）方法就新建了textarea（用来存放、编辑纯文本）、editorPane（用来存放、编辑Html文本）及scrollPane（用来包含上面的这些editor，当scrolling  的时候，cursor不可见）！！！！ [mxInteractiveCanvas](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\swing\view\mxInteractiveCanvas.html)：定义了一个canvas用来执行对graph的是否命中的检测 | |

# [com.mxgraph.util](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\package-summary.html) 包：

**该包提供了诸如mxConstants, mxUtils, mxPoint 和mxRectangle等用于自定义事件及取消历史的类**。

|  |
| --- |
| Interfaces   [*mxEventSource.mxIEventListener*](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxEventSource.mxIEventListener.html) ：定义监听事件的各种需要 [*mxUndoableEdit.mxUndoableChange*](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxUndoableEdit.mxUndoableChange.html)：定义取消操作的各种需要 |

|  |
| --- |
| Classes   [mxCellRenderer](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxCellRenderer.html) ：cell的呈现：如创建图形、HTML文件并画出cell [mxCellRenderer.CanvasFactory](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxCellRenderer.CanvasFactory.html) ：创建Canvas，当文档的内容、文档的大小被告之后，将初始化和创建Canvas分离。 [mxConstants](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxConstants.html) ：包含所有全局常量。：定义了*DEFAULT\_FONTSIZ*E= 11;！！！！ [mxEvent](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxEvent.html) ：包含所有（与事件有关的）全局常量 [mxEventObject](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxEventObject.html) ：对一些被命名的事件进行重新分派事件 [mxEventSource](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxEventSource.html) ：为监听事件源的对象定义需求的东西 [mxImage](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxImage.html) ：设置一个image的大小、来源 [mxLightweightTextPane](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxLightweightTextPane.html) ： [mxPoint](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxPoint.html) ：实现一个双精度的二维的点 [mxRectangle](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxRectangle.html) ：实现一个双精度的二维的矩形 [mxResources](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxResources.html) ：存取资源捆 [mxUndoableEdit](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxUndoableEdit.html) ：定义取消、恢复操作 [mxUndoManager](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxUndoManager.html) ：  [mxUtils](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\util\mxUtils.html)：提供了使用mxgraph的各种帮助方法：getfont、getcolor、getString等 [com.mxgraph.view](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\package-summary.html)包： |
| 实现graph的component。 Interfaces   [mxEdgeStyle.mxEdgeStyleFunction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxEdgeStyle.mxEdgeStyleFunction.html)：边的样式的函数  [mxGraph.mxICellVisitor](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxGraph.mxICellVisitor.html) ： [mxPerimeter.mxPerimeterFunction](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxPerimeter.mxPerimeterFunction.html) | |

|  |
| --- |
| Classes   [mxCellState](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxCellState.html) ：一个cell的目前状态 [mxEdgeStyle](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxEdgeStyle.html) ：提供各种各样边的样式 [mxGraph](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxGraph.html) ：用graph model和样式表去创建一个graph [mxGraphSelectionModel](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxGraphSelectionModel.html) ：实现一个graph所选择的model [mxGraphSelectionModel.mxSelectionChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxGraphSelectionModel.mxSelectionChange.html)  [mxGraphView](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxGraphView.html) ：实现graph的view，管理为孩子 [mxGraphView.mxCurrentRootChange](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxGraphView.mxCurrentRootChange.html)  [mxLayoutManager](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxLayoutManager.html) ：实现layout manager [mxMultiplicity](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxMultiplicity.html) ：多重 [mxPerimeter](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxPerimeter.html)  [mxSpaceManager](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxSpaceManager.html)  [mxStyleRegistry](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxStyleRegistry.html)  [mxStylesheet](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxStylesheet.html) ：定义一个graph中cell的初始样式，可用set mxConstants.DEFAULT\_FONTSIZE来设置cell中文本的大小！！！ [mxSwimlaneManager](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxSwimlaneManager.html)  [mxTemporaryCellStates](file:///C:\Documents%20and%20Settings\Administrator\桌面\学习\jgraphx\docs\api\com\mxgraph\view\mxTemporaryCellStates.html)：cell的临时状态 |

state.getView().getGraph().getModel();

state = graphComponent.getGraph().getView().getState(cell)

从中可以看出什么？

### Mxcellstate类：

Mxcellstate (mxGraphView view, Object cell, Map<String, Object> style)方法：

构造一个新对象，表示指定的视图中特定cell的当前状态。

**Parameters:**

**view** 包含state的graph视图.

**cell** 该state描绘的cell

**style** Array of key, value pairs that constitute the style.

# 有关label的问题：

Label可以在mxGraph.cellLabelChanged 中进行给定cell的lanbel值（value）的设置。这个过程中mxEvent.LABEL\_CHANGED被触发。Com.mxgraph.swing.view.mxCellEditor类