FlatScrollBar滚动条控件的使用

作者：张青松

目录

[1. FlatScrollBar滚动条 2](#_Toc428457289)

[1.1. 添加滚动条 3](#_Toc428457290)

[1.1.1. 普通方法添加滚动条 3](#_Toc428457291)

[1.1.2. Table与Tree添加滚动条 6](#_Toc428457292)

[1.2. Codeffine.jar包中其他有用的类 6](#_Toc428457293)

# FlatScrollBar滚动条

FlatScrollBar是一个第三方滚动条控件，在org.knime.workbench.core项目中引入的codeaffine.jar包中。如图：

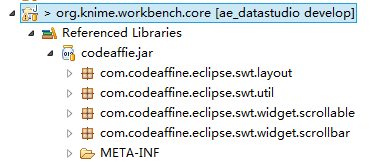


图1 codeaffine.jar包

使用FlatScrollBar可以在在任意位置添加滚动条，如图

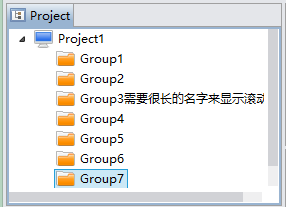


图2 下方和右边FlatScrollBar滚动条

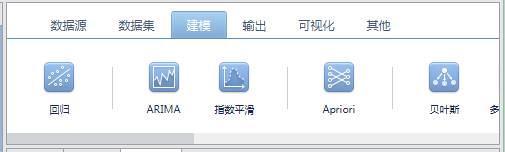


图3 下方的FlatScrollBar滚动条

## 添加滚动条

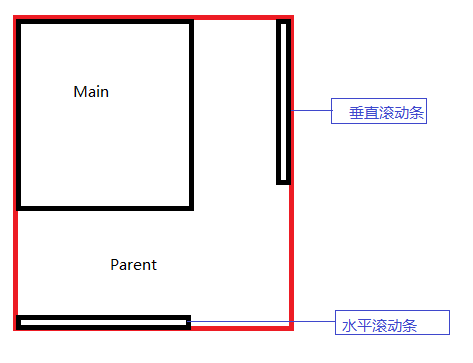
FlatScrollBar滚动条提供多种方式添加，可以如SWT的普通组件一样添加，针对Table和Tree控件，FlatScrollBar提供了现有的包装类com.codeaffine.eclipse.swt.widget.scrollableFlatScrollBarTable和com.codeaffine.eclipse.swt.widget.scrollable. FlatScrollBarTree类方便的添加滚动条。



### 普通方法添加滚动条

针对下图中的Main的类型选择滚动条的添加方式，对于简单的Composite等控件，选择这种方式，具体参考ModelView中滚动条的添加。

添加滚动条如图：



FlatScrollBar需要设置当前Main与Parent的Height和Weight的比例计算是否需要显示滚动条，以及滚动条的长度。

和其他SWT组件一样的方法初始化FlatScrollBar对象，并且将滚动条通过布局放到的Parent容器的对应位置。

FlatScrollBar vScroll = **new** FlatScrollBar(parent, SWT.*V\_SCROLL*);

FormData fd = **new** FormData();

fd.width = 16;

fd.height = parent.getSize().y;

fd.top = **new** FormAttachment(0, 0);

fd.right = **new** FormAttachment(100, 0);

fd.bottom = **new** FormAttachment(100, 0);

vScroll.setLayoutData(fd);

//当滚动条位置变化时，Main滚动到相应位置

vScroll.addSelectionListener(**new** SelectionAdapter() {

@Override

**public** **void** widgetSelected(SelectionEvent event) {

Control c = Mian;

Point location = c.getLocation();

c.setLocation(location.x, - vScroll.getSelection());

}

});

ControlListener listener = **new** FocusControlListener(Mian, vScroll);

//对Main控件添加监听，当Main的Size发生变化时，重新计算滚动条的长度

Mian.addControlListener(listener);

其中的监听Listener类如下：

**private** **class** FocusControlListener **implements** ControlListener {

**private** Control control;

// 解决滚动条对象不一致问题

**private** FlatScrollBar bar;

**public** FocusControlListener(Control control, FlatScrollBar bar) {

**this**.control = control;

**this**.bar = bar;

}

@Override

**public** **void** controlMoved(ControlEvent arg0) {

}

/\*\*

\* 设置滚动条

\*/

@Override

**public** **void** controlResized(ControlEvent arg0) {

**if** (bar != **null** && control != **null**) {

bar.setMaximum(control.getSize().y);// 设置需要滚动的控件总长度 bar.setThumb(control.getParent().getClientArea().height);// 设置能显示的控件宽度 bar.setPageIncrement(control.getParent().getClientArea().height); // 同上

control.setLocation(0, control.getLocation().y); // 显示pageContentComposite的最前端

bar.setSelection(0);// 使滚动条回到最前端

}

}

}

在Listener中通过滚动条的setMaximum()、setPageIncrement()和setPageIncrement()方法设置滚动条。其中setMaximum()传入Main的height，而setPageIncrement()与setPageIncrement()都传入Parent的height。

### Table与Tree添加滚动条

对于Main是Tree或者Table这样较为复杂的情况，提供了另一种封装好的类，位于com.codeaffine.eclipse.swt.widget.scrollable包中的FlatScrollBarTable和FlatScrollBarTree。

以FlatScrollBarTree举例说明使用方法。具体使用实例参见ExplorerView中滚动条的添加方式。

1. 继承ScrollableFactory重新create方法得到Tree对象

**public** **class** TreeScroFactory **implements** ScrollableFactory<Tree>{

@Override

**public** Tree create(Composite parent) {

// 同上生成Tree对象

TreeViewer m\_viewer = createTreeView(parent);

Tree tree = m\_viewer.getTree();

**return** tree;

}

}

1. 用FlatScrollBarTree对象关联Tree与Parent

ScrollableFactory<Tree> factory = **new** TreeScroFactory();

**new** FlatScrollBarTree(parent, factory);

针对其他Main组件是较为复杂的类型是，可以仿照FlatScrollBarTree自己的类。

## Codeffine.jar包中其他有用的类

在codeffine.Jar包中还有一个封装了FormLayout布局的工具类



通过FormDatas工具类可以方便的，以简短的代码完成FormLayout布局。使用举例：

FormDatas.*attach*(control\_A).atLeftTo(control\_B, 10, SWT.*TOP*)

.atTopTo(control\_C,12);