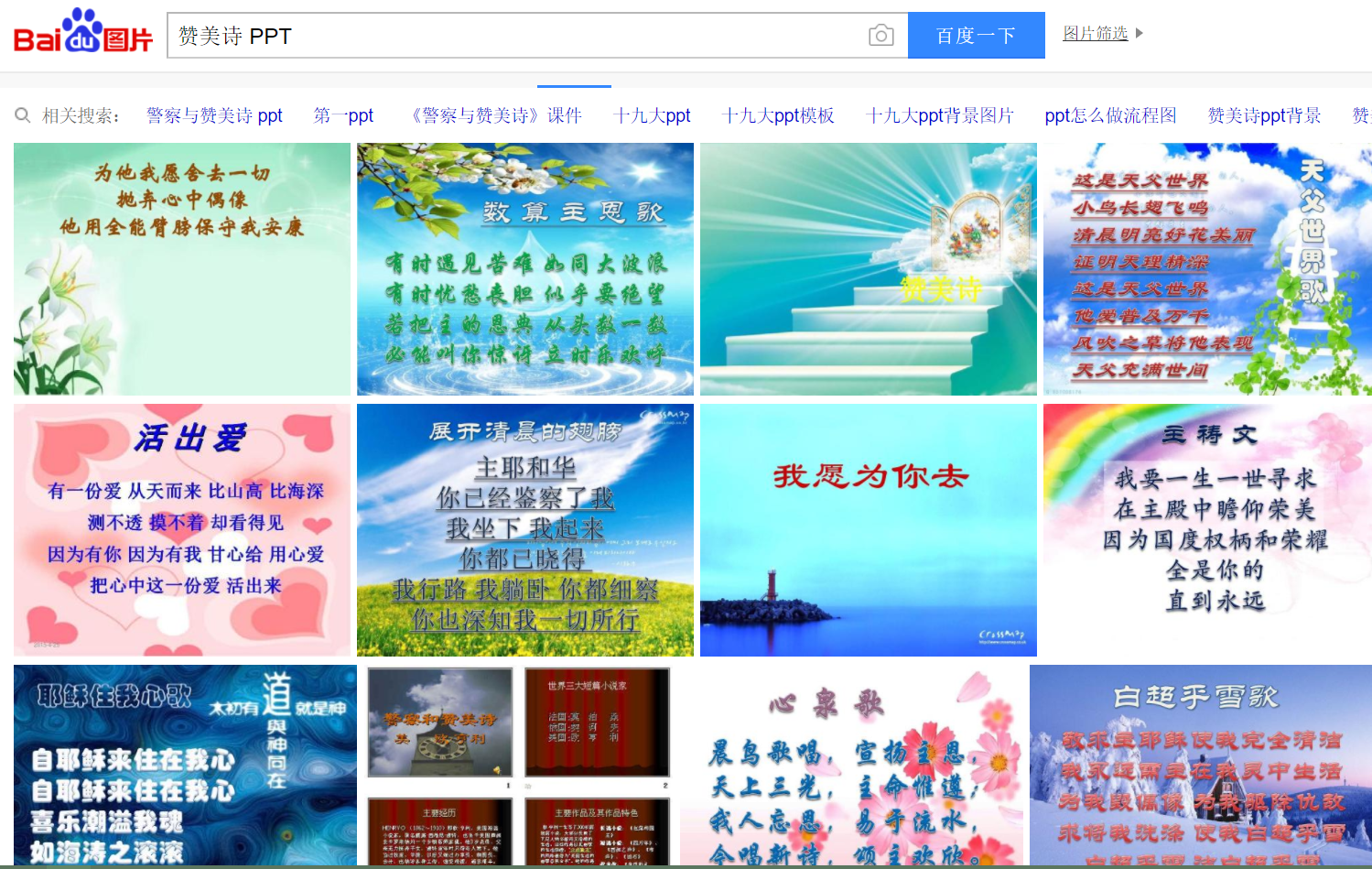
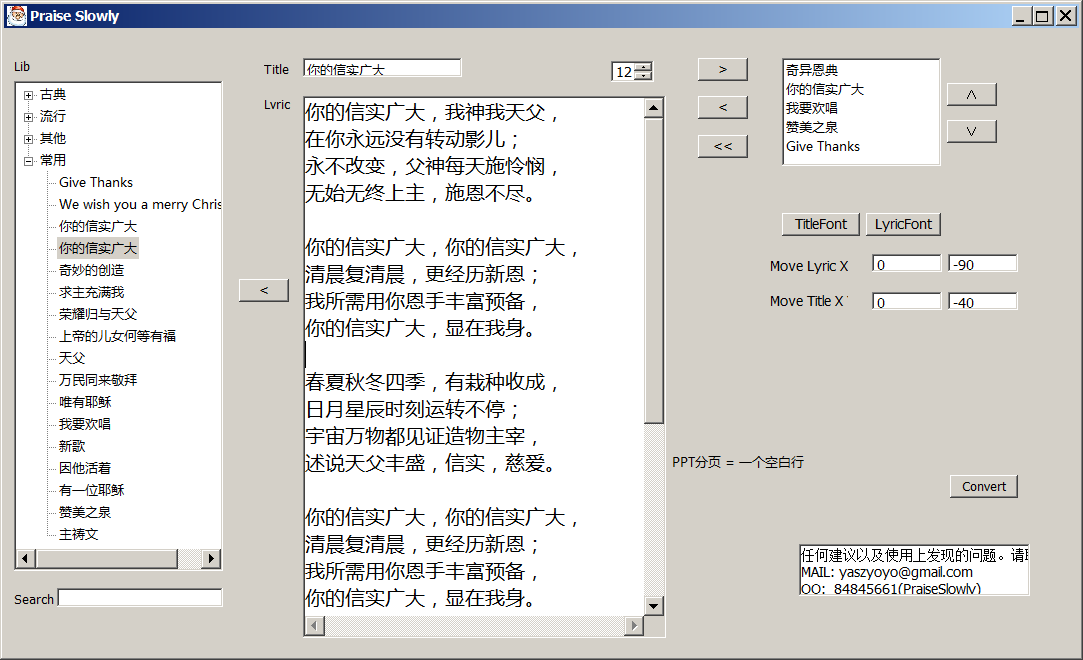
# 背景

从百度搜索赞美诗PPT可以看出目前教会的PPT，





praise slowly是一个歌词管理器+PPT生成器，目前java swt+poi开发的，加入快捷键以及批量歌词检索功能，后期预备开发：自定义模板，网络同步；

# 安装部署

环境：windows

安装jre-6u27-windows-i586.exe，执行run.bat（若无法启动，请编辑run.bat将JAVA路径写成绝对路径，路径要加引号）。

"C:\Program Files (x86)\Java\jre1.6.0\_27\bin\java.exe" -jar ps1.1.jar

nginx：

server {

listen 8887;

server\_name localhost;

location /api/ {

proxy\_pass http://localhost:8090/api/;

if ($request\_filename ~\* ^.\*?\.(txt|doc|pdf|rar|gz|zip|docx|exe|xlsx|ppt|pptx)$){

add\_header Content-Disposition: 'attachment;';

}

}

location / {

alias G:/dev/2.h5/3.vue/ps1.3ui/dist/;

try\_files $uri $uri/ /index.html;

}

location /search/ {

proxy\_pass https://api.zanmeishi.com/search/;

}

location /song/ {

proxy\_pass https://api.zanmeishi.com/song/;

}

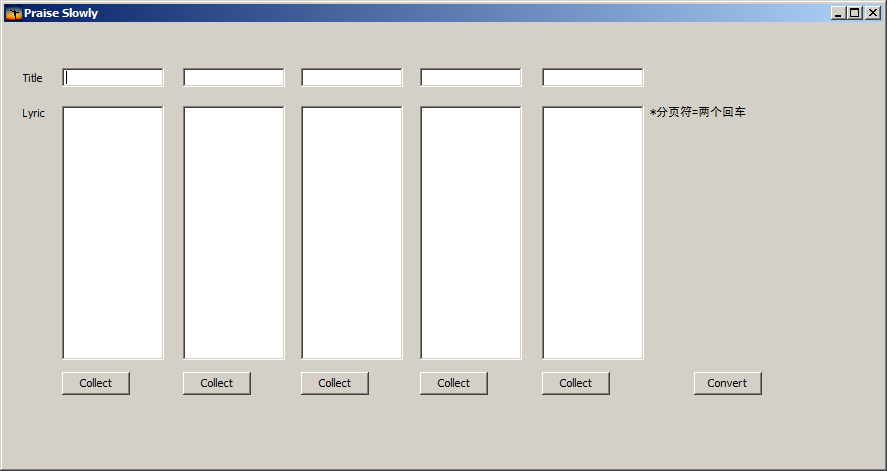
}

iis：web.config

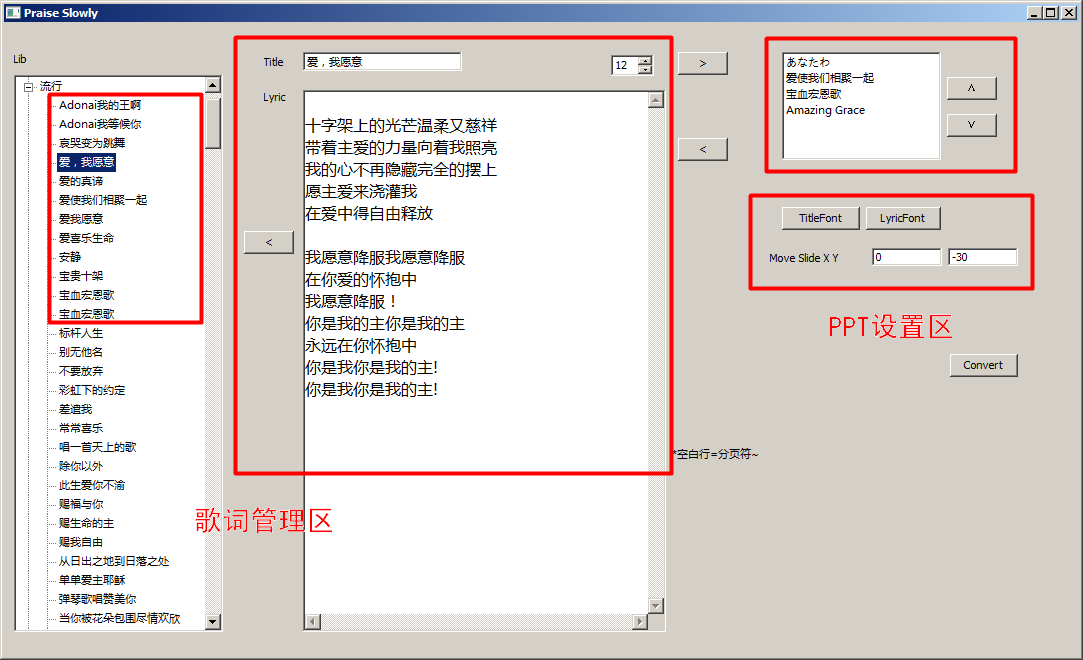
 <rewrite>  
            <rules>  
                    <rule name="RedirectToNginx" stopProcessing="true">  
                        <match url=".\*" />  
                        <action type="Rewrite" url="[http://ppt.zanmeishi.com:8887/{R:0](qq://txfile/)}" />  
                    </rule>  
                </rules>  
        </rewrite>

# 历史功能介绍

## v1

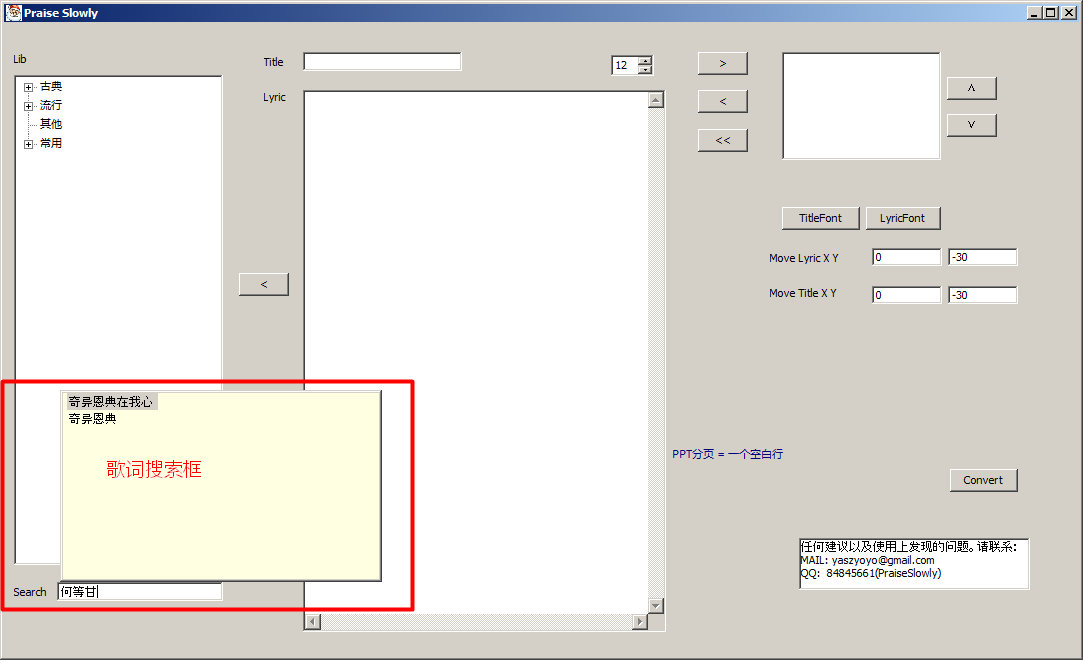


## v1.2



1. 基于固定模板生成PPT；中可设置title以及lyric字体、大小(特殊的下拉窗)、POSITION（上下左右距离）;
2. 歌词管理器自动保存新增歌词（srcType=0）；
3. 歌词模糊搜索(在如下Search框)

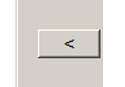
由于赞美诗重名的情况经常出现，如果只是通过title为KEY，不能一次定位到对应的诗歌，所以可以按歌词来搜索。



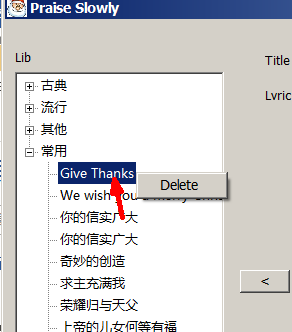
1. 歌词添加

增：

点击任意歌词，**先**修改标题，再修改文本，然后通过

将歌词导入到对应的目录下；

删：右键盘



## v1.3

praiseslowly 1.3

平均1分钟可导出5首诗歌ppt，使用手机/PC现代浏览器登录ppt.zanmeishi.com使用吧；

**使用说明:**  
\***歌词页**中搜索歌词或手动添加诗歌信息(以空行作为分页符，一次可导出5首);  
\***模板页**选择模板;  
\***导出页**选择好诗歌顺序，单击**导出**即可。

歌词搜索：使用zanmeishi api模糊检索歌词[首拼、带纠错功能]；



目前限于诗歌歌词质量，正则表达式处理：

|  |  |
| --- | --- |
| 单页行数太多 |  |
| 单行字数太多 |  |
| 带作者信息 | 词曲唱：佳偶  编曲：曹龙一  笛子:徐遥祝  录音师:费华为  字符数少于7个，但是包含:或者： |
| 单行太短 |  |

**模板导出功能:**

根据用户上传的PPT模板，将标题/歌词内容填充到ppt的中文本框中包含下列内容的文本框:

#title

#lyric

#order

### todolist

|  |  |
| --- | --- |
| 上下键拖动 2018.4 | <https://segmentfault.com/q/1010000006620125> |
| 后端模板管理 | 1.后端新增模板和截图：  root- webcontent-pic  root- webcontent-ppt  新增初始化程序将相应数据加载入redis，  2.前端template页面初始化请求后端截图和菜单信息， |
|  |  |

### 正则表达式

# CORE SRC

**SlideShow** \_slideShow = **new** SlideShow();

\_slideShow.setPageSize(**new** Dimension(pageWidth, pageHeight));

**Slide** \_masterSlide = **new** SlideShow(**new** HSLFSlideShow("1.ppt"))

.getSlides()[0];

**Slide** slide = \_slideShow.createSlide();

**for** (Shape sh : \_masterSlide.getShapes()) {

slide.addShape(sh);

}

## BUG LIST

通过手动（按歌曲类别、时间）分类(json)；必须先编辑标题，相关控件：

<http://hi.baidu.com/sonmeika/item/e50d67849fc06c2b100ef316>

输出歌词通过两个List数组，需要维护数组的移动、初始化加载；

### PPT最后一页只有一行的情况，歌词无法显示

故障分析：

\*回车有 (\r\n) 、(\n)两种模式；

\*POI无法在空文本后增加RTR；

### 当只有一页PPT，无法显示

故障分析：

**TextBox** tb = **new** TextBox();

**TextRun** tr = tb.createTextRun();

tr.setText("1");

tr.appendText("a").setFontSize(40);;

tr.appendText("b").setFontSize(32);

**RichTextRun** rtr = tr.getRichTextRuns()[0];

rtr.setText(" ");

### BUG 字体无法设置：

FONT:<http://hi.baidu.com/freish/item/3906ca81e1284bdfd0f8cd3a>

<http://blog.csdn.net/blognkliming/article/details/8680893>

PPFONT：

<http://www.oschina.net/code/explore/poi-3.7/src/scratchpad/testcases/org/apache/poi/hslf/model/TestPPFont.java>

TXT解析到PPT说明：

居中POSITION：通过设置满PPT宽度的单元格以及字体居中的方式；

\*歌词部分以两个回车作为分页符，兼容最后一行有一个回车(标黄处理)

### 拷贝BUG

Exception in thread "main" java.lang.UnsupportedOperationException: Adding a shape from a different container is not supported - create it from scratch witht XSLFSheet.create\* methods

插入到指定页码：

<http://wutaoo.iteye.com/blog/2155423>

默认SLIDE大小为720\*540：

tb.setAnchor(new Rectangle(360,270,0,0));

//前两个参数是BOX左上角的横向、纵向坐标（以左上角为原点，可以是负数），后面两个参数是BOX的长宽。另一个2参构造函数读入的是BOX的长宽。

参考：

杜磊米<http://282912533.iteye.com/blog/566798>

**TB能否动态增加文字：**

可以，

TB生成TextRun，

通过将动态增加RichTextRun到TextRun

一次设置字体：

tb.setText(lyric);//TB级

RichTextRun \_rtr = tb.getTextRun().getRichTextRunAt(0);//TB下两级

\_ rtr.setFontName("SimHei");

Tr.setText之后只能取1个；tr.getRichTextRuns()[0];

tr.appendText("hello")后，tr.getRichTextRuns()[1];

SWT另存功能<http://niub.iteye.com/blog/1787595>

<http://my.oschina.net/u/1269662/blog/192470>

 XWPFRun#setFontFamily()

<https://poi.apache.org/slideshow/ppt-file-format.html>

[How do I get background data from pptx slides using apache poi](http://stackoverflow.com/questions/18292287/how-do-i-get-background-data-from-pptx-slides-using-apache-poi)

#### 前后颜色FILL

<http://poi.apache.org/apidocs/org/apache/poi/hslf/model/Fill.html>

# SWT使用对象总结

\*PPT输出对象不应设置为ArrayList，而应直接使用SWT控件的Setdata()；

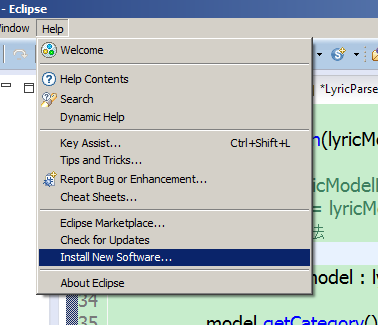
\*初始化数据的读取与析构函数代码在不同的位置(TreeListner与SWT窗体)；

Change Background of Slide Using Java

**import**org.apache.poi.hslf.HSLFSlideShow;  
**import**org.apache.poi.hslf.model.Slide;  
**import**org.apache.poi.hslf.usermodel.\*;  
**import**org.apache.poi.hslf.usermodel.SlideShow;  
**import**org.apache.poi.hslf.model.\*;  
**import**java.io.\*;  
**import**java.awt.Color;  
**import**org.apache.poi.hslf.model.TextBox;  
**class**changeBackground  
{  
  **public static void**main(String a[])  
  {  
  **try**  
  {    
  SlideShow slideShow = **new**SlideShow();  
  Slide slide = slideShow.createSlide();  
  Shape shape = **new**AutoShape(ShapeTypes.Rectangle);  
  shape.setAnchor(**new**java.awt.Rectangle(0, 0, 800, 660));  
  Fill fill = shape.getFill();  
  fill.setFillType(Fill.FILL\_SHADE);  
  fill.setBackgroundColor(Color.blue);  
  fill.setForegroundColor(Color.green);    
  slide.addShape(shape);  
  FileOutputStream out = **new**FileOutputStream  
("changeBackground.ppt");  
  slideShow.write(out);  
  out.close();   
  }**catch**(Exception e){}  
  }}

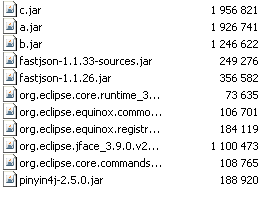
## 环境构建

Eclipse下，安装SWT开发插件:



WindowBuilder - <http://download.eclipse.org/windowbuilder/WB/release/R201309271200/4.3/>

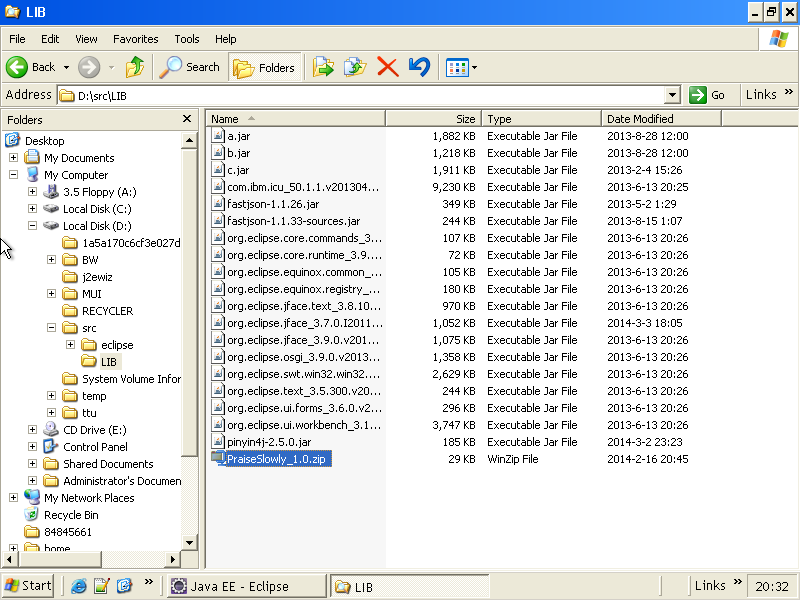
## 部署



11/11

D:\src\1.SRC\swt-4.3-win32-win32-x86\_64\output\lib

其中a,b,c是 32bit下的POI、 SWT



使用J2EWIZ.rar部署

## PSTreeListener.java 负责加载-> LyricParser.parseTreeByJsonFille

## 代码片段

### SWT TREE

#### 寻找INDEX位置排序

**public** **class** **TICompare** **implements** Comparable<**TICompare**>{

**public** **TreeItem** ti;

**public** TICompare(**TreeItem** ti){

**this**.ti = ti;

}

**public** **int** compareTo(**TICompare** tc) {

// **TODO** Auto-generated method stub

**return** Collator.*getInstance*(**Locale**.*CHINESE*).compare(**this**.ti.getText(),tc.ti.getText());

}

}

实现Comparable的类可以通过Arrays.sort(tiCompares)进行排序，

http://www.java2s.com/Code/Java/SWT-JFace-Eclipse/DragandDropexamplesnippetdragleafitemsinatree.htm

### SWTModel

TextBox tb = new TextBox();

String tair = lineStrings[lineStrings.length - 1];

TextRun \_textRun = tb.createTextRun();//创建了

RichTextRun \_richTextFront = \_textRun.getRichTextRuns()[0];

\_richTextFront.setText(front);

\_richTextFront.setFontName(fontStr);

\_richTextFront.setFontSize(fontSize);

\_richTextFront.setFontColor(Color.white);

\_richTextFront.setAlignment(1);// 居中

\_textRun.appendText(tair);//不能放在后两行，要先有字TR，后有RTR

RichTextRun \_richTextTair = \_textRun.getRichTextRuns()[1];

\_richTextTair.setFontName(fontStr);

\_richTextTair.setFontSize(fontSize);

\_richTextTair.setFontColor(Color.yellow);

\_richTextTair.setAlignment(1);// 居中

### 首字母搜索

<http://www.boyunjian.com/do/article/snapshot.do?uid=net.csdn.blog%2Fyawolf%2Farticle%2Fdetails%2F2676626>

<http://lggege.iteye.com/blog/268306>

<http://bbs.csdn.net/topics/350080624>

a)ContentProposalAdapter

i.内容建议Adapter类

b)IControlContentAdapter

i.内容接口

c)TextContentAdapter,ComboContentAdapter

i.内容Adapter类[专用于Text或Combo的].

ii.实现了IControlContentAdapter.

<http://stackoverflow.com/questions/13313410/is-eclipse-autocompletefield-only-suggesting-entries-starting-with-entered-text>

d)SimpleContentProposalProvider

i.建议Provider类.

e)AutoCompleteField

i.从ContentProposalAdapter和SimpleContentProposalProvider扩展出来的一个简易使用的类.

Adapter

按照帮助文档的描述如下：为了在用户从弹出窗口中选择建议时获取和设置控件内容，必须向适配器提供[**IControlContentAdapter**](http://127.0.0.1:62318/help/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/jface/fieldassist/IControlContentAdapter.html) 实例，该实例可以检索和设置特定类型控件的内容。对于文本字段来说，可以使用[**TextContentAdapter**](http://127.0.0.1:62318/help/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/jface/fieldassist/TextContentAdapter.html) 类。但是，也可以灵活地实现 [**IControlContentAdapter**](http://127.0.0.1:62318/help/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/jface/fieldassist/IControlContentAdapter.html)，以便将内容建议适配器与任何其他类型的控件配合使用。  
    关键是定义**ContentProposalAdapter类，**它有几个必须的参数，第一个是需要进行内容辅助的控件，第二个是[**IControlContentAdapter**](http://127.0.0.1:62318/help/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/jface/fieldassist/IControlContentAdapter.html) 实例，jface默认提供了二个实现：[ComboContentAdapter](http://127.0.0.1:62318/help/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/jface/fieldassist/ComboContentAdapter.html),[TextContentAdapter](http://127.0.0.1:62318/help/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/jface/fieldassist/TextContentAdapter.html)。按照上面的建议，如果你需要更加高级的功能，那么得实现[**IControlContentAdapter**](http://127.0.0.1:62318/help/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/jface/fieldassist/IControlContentAdapter.html)**。**第四个是**IContentProposalProvider**的实例，这是提供内容辅助窗口中的内容提供器的接口，jface只实现了一个**SimpleContentProposalProvider**，这是最简单的只提供文本内容的内容提供器。同样的，你可以继承**IContentProposalProvider**来实现更高级的功能。  
    keyStroke是定义按下什么键时激活该提示。autoActivationCharacters是定义当输入遇到什么符号时会激活提示的字符数组。  
   按照帮助文档的说明，可以在任意控件上安装 [**ContentProposalAdapter**](http://127.0.0.1:62318/help/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/jface/fieldassist/ContentProposalAdapter.html) 以提供此行为。需要什么样的高级功能，就必须具体实践一下以上几个接口的实现。一般的编辑器都会配合SoucreViewer来提供这些功能。

// Create the find content assist field

final TextContentAdapter contentAdapter = new TextContentAdapter(); // 文本内容适配器

final IContentProposalProvider findProposer = new XzqhContentProposalProvider(); // 行政区划代码内容建议供应者

final ContentAssistCommandAdapter fContentAssistFindField = new ContentAssistCommandAdapter(

cbcRegional, contentAdapter, findProposer,

ContentAssistCommandAdapter.CONTENT\_PROPOSAL\_COMMAND,

IContentAssist.AUTO\_ACTIVATION\_CHARACTERS, false);

fContentAssistFindField

.setProposalAcceptanceStyle(ContentProposalAdapter.PROPOSAL\_REPLACE);

