开源GIS调研

# 概述

两种实现方式：1、直接在控件中导入矢量数据，显示地图；2、通过网上地图API，调用URL获得网站地图，显示在控件中。由于无法连外网，第二种方法可利用Map Server和Geo Server搭建地图服务器实现局域网内获得地图。

## 方法1

Geotools作为开源GIS软件中地图显示处理的工具，可直接用于开发显示地图的控件。或者通过利用开源GIS软件中已实现的控件，直接显示地图。

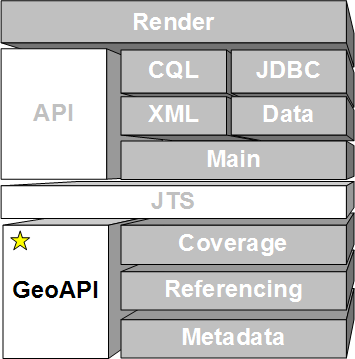
### Geotools

大多数开源GIS软件，如OpenJump，uDig，MyMaps都基于Geotools开发。提供了丰富API，包括返回地图图像，读写各种地理数据类，如矢量、栅格等。

背景：

Geotools是Java语言编写的开源GIS工具包。该项目已有十多年历史，生命力旺盛，代码非常丰富，包含多个开源GIS项目，并且基于标准的GIS接口。Geotools主要提供各种GIS算法，各种数据格式的读写和显示。在显示方面要差一些，只是用Swing实现了地图的简单查看和操作。但是用户可以根据Geotools提供的算法自己实现地图的可视化。

Geotools用到的两个较重要的开源GIS工具包是JTS和GeoAPI。前者主要是实现各种GIS拓扑算法，也是基于GeoAPI的。但是由于两个工具包的GeoAPI分别采用不同的Java代码实现，所以在使用时需要相互转化。Geotools又根据两者定义了部分自己的GeoAPI，所以代码显得臃肿，有时容易混淆。由于GeoAPI进展缓慢，Geotools自己对其进行了扩充。另外，Geotools现在还只是基于2D图形的，缺乏对3D空间数据算法和显示的支持。



### MyMaps

**关键字**：swing，开源，只提供简单的浏览缩放功能

**主页**：<http://sourceforge.net/projects/mymaps/>

**源码**：cvs -d:pserver:anonymous@mymaps.cvs.sourceforge.net:/cvsroot/mymaps login

cvs -z3 -d:pserver:anonymous@mymaps.cvs.sourceforge.net:/cvsroot/mymaps co -P myMaps

最后更新时间为2009年，根据user manuel的步骤操作可运行成功

### OpenJUMP

**关键字**：SWING，开源，是开源项目JUMP的后续开发版本

**官网**：<http://openjump.org/>

**源码**：

svn co [https://jump-pilot.svn.sourceforge.net/svnroot/jump-pilot/core/tags/1.5.2 openjump-core-1.5.2](https://jump-pilot.svn.sourceforge.net/svnroot/jump-pilot/core/tags/1.5.2%20openjump-core-1.5.2)

### UDig

**关键字**：基于RCP和Geotools开发的插件项目，功能较丰富，根据网上步骤工程跑不起来？

**官网**：<http://udig.refractions.net/>

### QGIS

关键字：C++，代码完整

## 方法2

地图API主要调研了google map api，具体采用何种API，具体情况具体分析。Java应用程序中用于显示google地图swing控件为DJ NativeSwing，JMapViewer和JXMapViewer为显示其他地图的swing控件。

### Google Map API

[**主流地图API比较**](http://www.iteye.com/topic/566712)**：**<http://www.iteye.com/topic/566712>

Google Map api开发快速入门：<http://wenku.baidu.com/view/704d821ea76e58fafab003b0.html>

GMapWTP，google map api代码补全插件<http://www.langtags.com/gmapwtp/index.html>

将google map api嵌入java 应用程序

<http://java-buddy.blogspot.se/2012/03/embed-google-maps-in-javafx-webview.html>

### JMapViewer

**关键字**：JMapViewer与JXMapViewer类似，但更简单，而且没有采用swingx控件，包含一个桌面程序的例子。由OpenStreetMap开发人员根据JOSM开发。

**官网**：<http://wiki.openstreetmap.org/wiki/JMapViewer>

<http://today.java.net/article/2007/10/24/building-maps-your-swing-application-jxmapviewer#building-a-basic-application>

**源码**：svn http://svn.openstreetmap.org/applications/viewer/jmapviewer/releases

<http://wiki.openstreetmap.org/wiki/JMapViewer>

下载后导入Eclipse即可运行成功

### JXMapViewer

**关键字**：JXMapViewer是一个开源swing控件，提供一个在java应用中显示地图的JPanel，载入和显示栅格，采用swingx控件。

**主页**：<http://wiki.openstreetmap.org/wiki/JXMapViewer>

**源码**：

<http://www.jarvana.com/jarvana/view/org/swinglabs/swingx-ws/1.0/swingx-ws-1.0.jar!/org/jdesktop/swingx/JXMapViewer.class?classDetails=ok>

只提供jar包和src源码，无开发环境和手册

# 其他

## 开源GIS网站

http://freegis.org

<http://opensourcegis.org/>

最主要的开源GIS服务器：Map Server和Geo Server，用于地图发布，综合性能Map Server强于Geo Server。

## 我用过的几个开源GIS软件

从空间数据库说起吧。在空间数据库领域，商业化的产品有ORACLE SPATIAL，以及以前的ARC SDE等等，而开源空间数据库虽然在功能上略有欠缺，但是在性能上毫不逊色。开源的空间数据库主要是postgresql和mysql的空间插件，分别是postGIS和mySpatial。目前由于PG对于面向对象支持的更好一些，所以postGIS在性能上和功能上都比myspatial要强一些。但是两者都有一个最重要的缺陷，就是不直接支持栅格数据。解决方案是将栅格数据存储为BLOB类型，并且对其建立四叉树索引，用以模拟商业数据库对栅格数据的存储。相对来说我更喜欢POSTGIS一些，原因有三：1，MYSQL本身不是在所有情况下免费，而PG在所有情况下都免费；2,POSTGIS性能和功能上都强于myspatial;3,国外已经有很多基于POSTGIS的成功应用，可以降低风险。

接下来轮到桌面软件了。QGIS和GRASS是很好的桌面GIS，当然基于ECLIPSE平台的UDIG也是另外一个选择。QGIS的最大特点在于界面很友好，熟悉ARCGIS的人都能很快的掌握QGIS的操作，另外QGIS对WMS的支持也不错。再者QGIS可以无缝集成POSTGIS，最后QGIS几乎完全照搬了GRASS的分析功能，因此其分析功能也很强大。GRASS本是美国军方开发的GIS，开源后一直受到美国大学老师们的青睐，它可以提供很多ARCGIS能提供的分析功能（当然每一种功能可选的算法没后者多），但是GRASS的界面比较丑陋，很多功能需要手动运行命令，因此不太适合大多数人用。基于eclipse平台的UDIG是桌面GIS的又一个选择，由于它是基于eclipse平台的，运行起来比较慢一点，而且比较耗内存。与前两者相比，它的优势在于操作的简单性，而且支持很多种空间数据源，包括很多商业空间数据库，缺点主要是分析功能比较弱。不过现在UDIG发展很迅速，也建议尝试一下。

开源GIS服务器主要有map server和geo server。Map server有两个版本，完全免费的那个那个版本是由明尼苏达大学开发的，全部是用C语言完成，效率很高可以媲美ARCGIS SERVER，并且支持我所知道的所有GIS数据源，但是相对来说简陋一些，并且没有内置AJAX支持。而map server的企业版本来是auto desk公司的map guide,这个版本功能很强大，而且还有IDE支持，当然IDE也有两个版本（付费的和不付费的），此外auto desk还封装了很多AJAX组建，因此这个版本可以说是真的企业级支持了。而geo server则是另外的选择了，

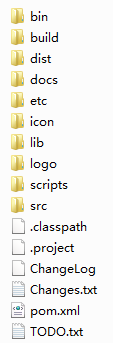
它是基于JAVA平台做的，安装时需要JDK1.4（高版本的也不行），其功能上和map server的完全免费版类似，但是性能上次之，只是如果对跨平台要求比较高的话，可以考虑使用它。此外，OTN上的文章一直喜欢使用GEO SERVER来发布ORACLE SPATIAL的地理数据，也不知道是为什么。

# OpenJUMP源码分析

## 安装步骤

<http://sourceforge.net/apps/mediawiki/jump-pilot/index.php?title=FAQ_OpenJump_and_Eclipse>

下载源码后，新建文件夹，取名OpenJUMP，在OpenJUMP目录下新建以下文件夹:



* **\build** => here the class files from the ant built will be stored (note: you may create the folder by yourself if not contained in the download)
* **\dist** => after building you will find here you distribution files (note: you may create the folder by yourself if not contained in the download)
* **\docs** => is currently empty but will later contain some pdf files (note: you may create the folder by yourself if not contained in the download)
* **\etc** => this must contains several files including the built.xml file for ANT (take the folder from the download)
* **\lib** => contains all the necessary libraries (including 2 subfolders: \Batik and \ext) (take the folder from the download, the BATIK folder may not be there in newer versions)
* **\scripts** => should contain the openJUMP start-up scripts for different OS (take the folder from the download)
* **\src** => includes the Jump\OpenJUMP source files needed to built OpenJUMP. It should contain at least 4 subfolders:

\src\com

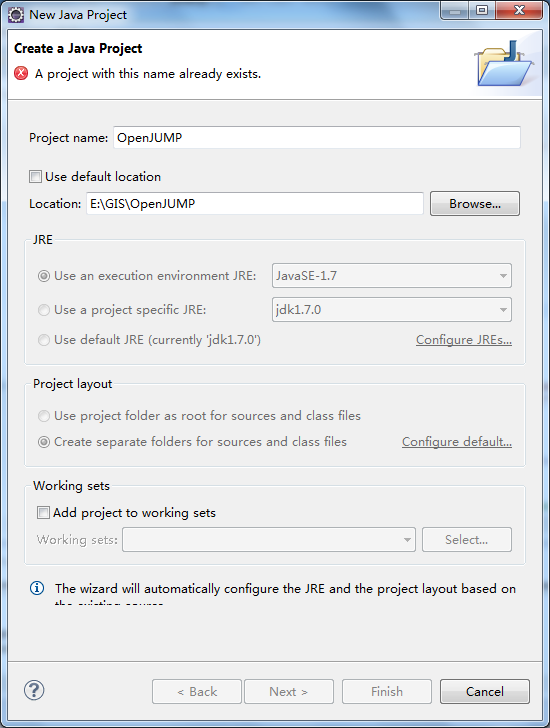
\src\org

\src\de

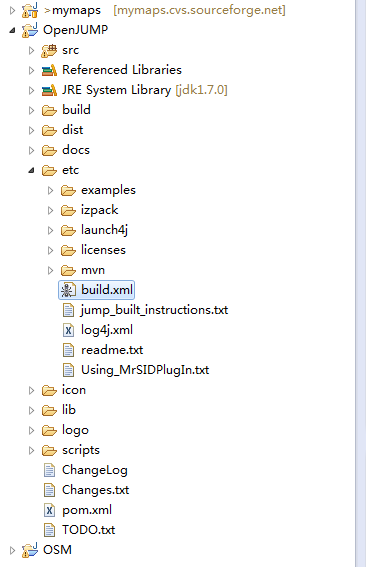
\src\language,

其中docs，etc，icon，lib，logo，script，src目录，和ChangeLog，Changes.txt，TODO，pom.xml中源码中拷来，其余目录为新建。

Eclipse新建工程，取名OpenJUMP，在Use default location中选择OpenJUMP目录的路径，Finish。



在OpenJUMP项目中etc->build.xml，右键Run As->ant build，则看到Build Success，则编译成功。



主类：com.vividsolutions.jump.workbench.JUMPWorkbench

程序参数:

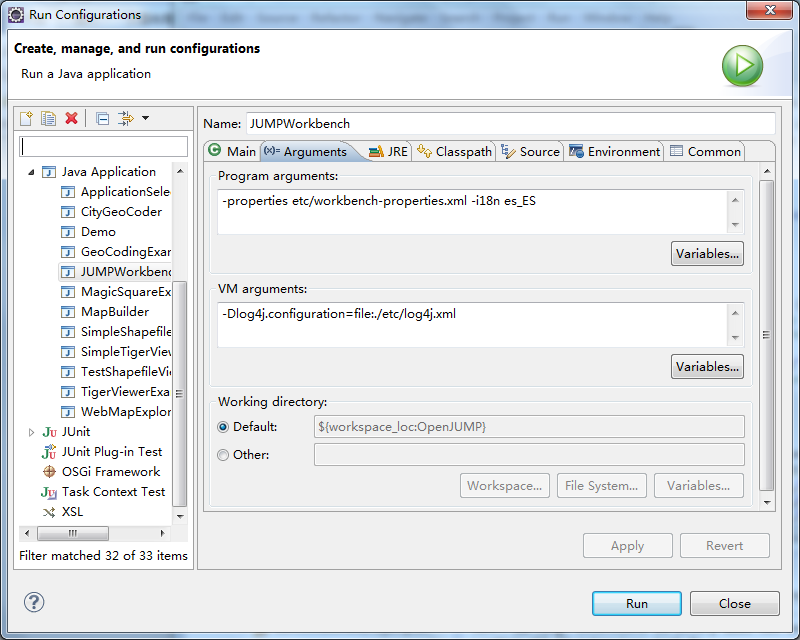
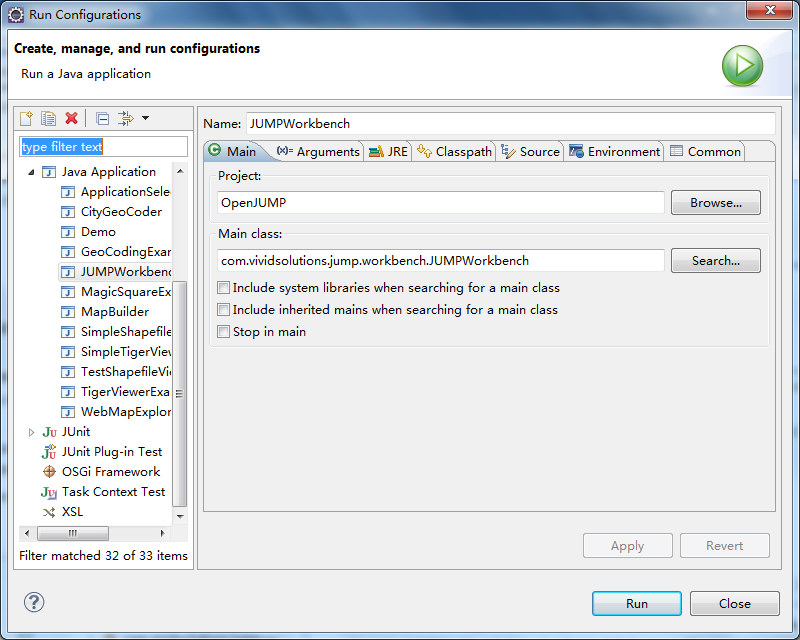
-properties etc/workbench-properties.xml 采用其他插件

-i18n：采用何种语言

-i18n lang\_country) (lang\_country :language 2 letters + "\_" + country 2 letters)

VM虚拟机参数: -Dlog4j.configuration=file:./etc/log4j.xml

在菜单栏->Run->Run Configuration->配置Main class，Program arguments，VM arguments，单击Run



## Plugin and Extension

OpenJUMP源码由2部分com和org包组成：

* com.vividsolutions.jump： 包含原来JUMP的代码，其中的类可以修改，但新增的功能要放入org.openjump 包中。
* org.openjump.core ：JUMP开发结束后，所有新增内容都放入该包中。新插件在org.openjump.core.ui.plugin.xxx包中。

如何构造一个plugin：

<http://sourceforge.net/apps/mediawiki/jump-pilot/index.php?title=How_to_use_and_make_own_Plugins>

XXPlugin继承自AbstractPlugIn，需实现两个方法initialize和execute，initialize负责界面初始化工作，例如在菜单栏中增加XXPlugin的菜单项，execute负责实现单击菜单项后的具体操作。其为线程类型，以保证在程序长时间执行时GUI失去响应，查看ThreadBasePlugin。

XXExtension继承自Extension，负责将Plugin注册到GUI中。在extension的configure方法中调用plugin的initialize，完成界面初始化工作。Extension可注册多个plugin的类。

**发布Plugin：**

1. 将项目打成jar包后放在lib/ext中；
2. 由两种方式将Plugin发布
3. 方式一：不推荐，在etc下新建workbench-properties.xml文件，输入XXPlugin的包+类名；

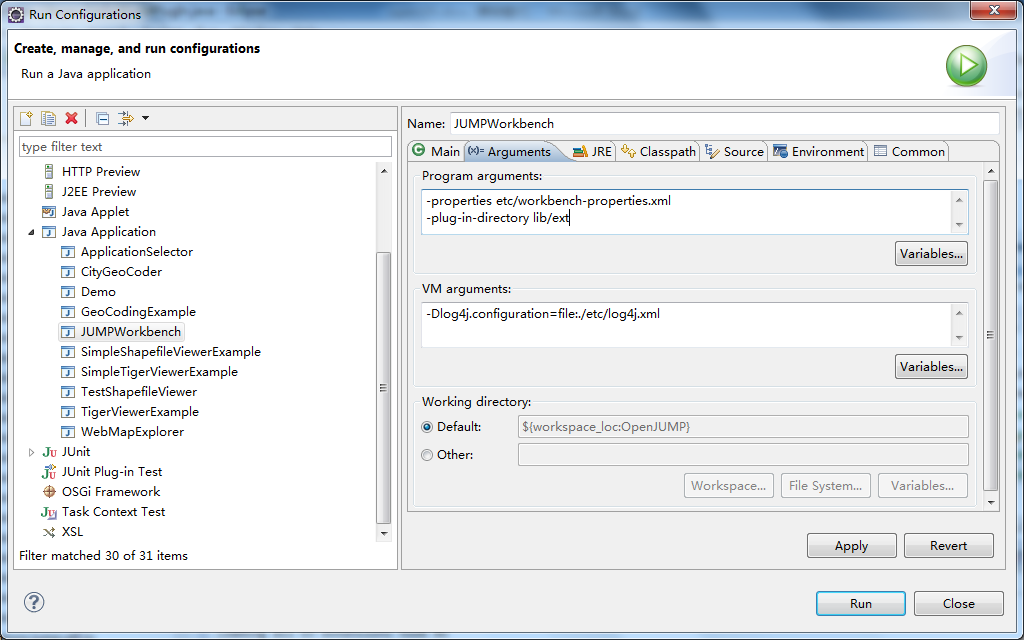
<workbench>

<plug-in>ch.unizh.geo.degen.HelloWorldPlugIn</plug-in>

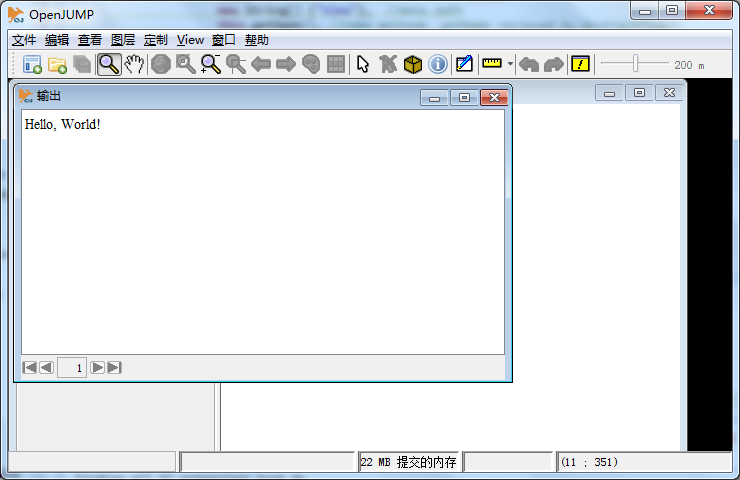
</workbench>

然后在Run As->Run Configuration中指定-properties etc/workbench-properties.xml

1. 方式二：在Run As->Run Configuration中指定-plug-in-directory lib/ext指定新增插件放置的目录；或在源码中查找Warning: Extensions directory does not exist，修改该处extensionsDirectory = new File("**lib/ext**"); 则不用指定-plug-in-directory。



1. 运行



## 命令行解析

CommandLine：命令行类

OptionSpec：命令行参数类

Option：解析结果类，由name和args组成，即参数和值

String I18N\_FILE=”i18n”; //参数名

CommandLine commandLine = new CommandLine('-'); //新建命令行对象，参数前缀为”-”

commandLine.addOptionSpec(new OptionSpec(I18N\_FILE, 1)); //新增需解析的参数，该参数的值只能出现1次

commandLine.parse(args); //解析命令行参数

if (commandLine.hasOption(I18N\_FILE)) { //判断命令行是否有该参数

I18N\_SETLOCALE = commandLine.getOption(I18N\_FILE).getArg(0); //获得第一个参数值

// initialize I18N

I18N.loadFile(I18N\_SETLOCALE); //载入该文件

}

## I18n

I18n用于设置界面字体的单例类

首先在src/language下新增一个\*\*.properties文件，如 jump\_zh\_CN.properties，其命名格式为jump\_language\_country,将jump.properties文件内容copy过来，修改其中对应的字符。

然后通过在Run As->Run Configurations->Java Application->Arguments->Program Arguments中增加-i18n 字体代号，为界面设置相应字体。

如果I18N\_SETLOCALE为空，则由*jumpResourceBundle*获得本地语系和地区

I18N.loadFile(I18N\_SETLOCALE); //载入对应的参数文件

I18N.get("ui.AboutDialog.version"); //获得ui.AboutDialog.version对应的字体

## Java2XML/ XML2Java

Java2XML将Object写入XML文件的类

Java2XML java2Xml = new Java2XML(); //新建Java2XML类

java2Xml.write(layer, "layer", inputXML); //layer退化成Object，”layer”是根节点名称，inputXML是File类，即XML文件

## 应用程序保存与恢复

JUMPWorkbench中

main(args, I18N.get("JUMPWorkbench.jump"), newJUMPConfiguration(),

splashPanel, progressMonitor); //构造JUMP配置，其中调用persistentBlackboard的initialize方法

PersistentBlackboardPlugIn persistentBlackboard = new PersistentBlackboardPlugIn();

persistentBlackboard.initialize(workbenchContext.createPlugInContext());

//其中restoreState(context.getWorkbenchContext());//载入./workbench-state.xml文件

WorkbenchFrame中

restore方法调用recallMaximizedState，recallWindowSize，recallWindowLocation方法获得上次关闭应用时的状态

saveWindowState在exitApplication中调用用于保存当前应用程序状态

## 第一个项目生成

JUMPWorkbench：

main(args, I18N.get("JUMPWorkbench.jump"), newJUMPConfiguration(), splashPanel, progressMonitor);

JUMPConfiguration：

private NewTaskPlugIn newTaskPlugIn = new NewTaskPlugIn(); //新建项目对象

NewTaskPlugIn：

public boolean execute(PlugInContext context) throws Exception {

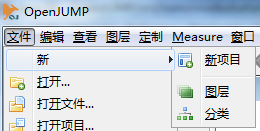
reportNothingToUndoYet(context);

context.getWorkbenchFrame().addTaskFrame(); //调用WorkbenchFrame中的addTaskFrame方法，即可新增项目

return true;

}

## 菜单项



JUMPConfiguration文件中：

//新建插件

NewTaskPlugIn newTaskPlugIn = new NewTaskPlugIn();

//新建插件生成对象

FeatureInstaller featureInstaller = new FeatureInstaller(workbenchContext);

//新建插件是否可用对象

EnableCheckFactory checkFactory = new EnableCheckFactory( workbenchContext);

//在”文件”菜单项下，新建”新”菜单项

FeatureInstaller.addMainMenu(featureInstaller, new String[] {MenuNames.FILE }, MenuNames.FILE\_NEW, 0);

//在”新”菜单项下新增”项目”菜单项

featureInstaller.addMainMenuItem(newTaskPlugIn, new String[] { MenuNames.FILE, MenuNames.FILE\_NEW }, newTaskPlugIn.getName(), false, newTaskPlugIn.getIcon2(), null);

//在”新”菜单项下新增分隔线

featureInstaller.addMenuSeparator(new String[] {MenuNames.FILE, MenuNames.FILE\_NEW});

//在”新”菜单项下新增”图层”菜单项

featureInstaller.addMainMenuItem(addNewLayerPlugIn,new String[] {MenuNames.FILE, MenuNames.FILE\_NEW}, new JMenuItem(I18N.get("com.vividsolutions.jump.workbench.ui.plugin.AddNewLayerPlugIn.name"), IconLoader.icon("layers.png")), checkFactory.createWindowWithLayerViewPanelMustBeActiveCheck());

EnableCheckFactory：判断该菜单项或工具项是否可用类

createWindowWithLayerViewPanelMustBeActiveCheck：当前活动窗口为LayerManagerProxy时，控件可用

createTaskWindowMustBeActiveCheck：当前活动窗口为TaskFrame时，控件可用

在LayerViewerPanel上加图形，参考比例尺

在LayerViewerPanel上左键单击事件，参考zoomIn按钮CursorTool

在LayerViewerPanel上添加Tooltip，参考ToolTipWriter，MapToolTipsPlugIn

<http://sourceforge.net/apps/mediawiki/jump-pilot/index.php?title=Working_with_Databases>

打成Runnable JAR file

## 类层次

图层类：

Layerable->

AbstractLayerable->

{Layer, WMSLayer, RasterImageLayer} //具体类包括普通图层，WMS图层，栅格图层

工具类：

CursorTool->

AbstractCursorTool->

DragTool->

AbstractZoomTool->

{PanTool, ZoomRealtimeTool, ZoomTool} //具体类包括浏览工具，缩放工具

CursorTool->

DelegatingTool->

QuasimodeTool //当同时按下shift/ctrl键时的工具

插件类:

PlugIn->

AbstractPlugIn

# 问题

## java.lang.SecurityException解决办法

Exception in thread "main" java.lang.SecurityException: invalid SHA1 signature file digest for com/microsoft/sqlserver/jdbc/SQLServerException.class  
插件打成的jar包运行出现这个异常，异常处理办法

1、把源sqljdbc.jar驱动解压到sqljdbc目录

2、删除 sqljdbc\META-INF\目录中出mainfest.mf，zigbert.rsa 以外的文件。

3、修改 sqljdbc\META-INF\目录中的mainfest.mf ，保留前三行，其余删除

4、重新打包：进入sqljdbc目录,执行:jar cvf sqljdbc.jar . 注意这最后的一个点号不能掉

5、将新打包的文件加入项目工程中即可。

Jar包内的MANIFEST.MF内容如下：

Manifest-Version: 1.0

Class-Path: .

Main-Class: examples.GISWorkbenchLoad

Created-By: 1.7.0 (Oracle Corporation)

## Unsupported major.minor version 51.0解决方法

由于使用jar包的项目的jdk版本，低于编译jar的jdk版本，会包该错

打包项目->右键properties->java compiler->configure workspace setting->compiler compliance setting，重新设置jdk level与运行项目一致。

## Premature end of file解决方法

在对XML内容进行解析的时候，比如用Digester将XML内容转化为Java对象时，有时会出现错误：

org.xml.sax.SAXParseException: Premature end of file

通常有两种情况会导致这种错误：

第一种：被解析的内容为空字符串

第二种：被解析的内容只有一行如：<?xml version=”1.0″ encoding=”UTF-8″?>

示例代码片断如下：

InputStream inputObject =new ByteArrayInputStream(content.getBytes(decode));

//报错行，其中content就是上述两种情况之一  
digesterObj.parse(inputObject);

## yum停在Setting up Install Process 解决方法

rm –f /var/lib/rpm/\_\_db\*

## running yum-complete-transaction first解决方法

yum install yum-utils

yum-complete-transaction -cleanup-only

package-cleanup –dupes

package-cleanup --problems

# Flash控件

## JFlashPlayer

FlashPanel一套通过调用dll显示Flash的swing控件

只支持win，不支持Linux，目前已经不再维护

## DJ NativeSwing

JFlashPlayer一套可显示Web Browser, Flash Player, HTML Editor, Multimedia Player的swing控件

官网说支持win,linux,mac，但linux实验不成功，且应用关闭时经常报错，用DJSweet不会报错。

Yum install xulrunner firefox

ln -s -t /usr/lib/xulrunner/libflashplayer.so /usr/lib/mozilla/plugins/libflashplayer.so

java -Dorg.eclipse.swt.browser.DefaultType=mozilla -Dorg.eclipse.swt.browser.XULRunnerPath=/usr/lib/xulrunner -jar test.jar

安装mozilla和xulrunner

xulrunner --register-global

String xulHome = **new** File("lib/xulrunner").getAbsolutePath();

NSSystemPropertySWT.WEBBROWSER\_XULRUNNER\_HOME.set(xulHome);

System.setProperty("org.eclipse.swt.browser.XULRunnerPath", xulHome);

-Dorg.eclipse.swt.browser.DefaultType=webkit -Dorg.eclipse.swt.browser.UseWebKitGTK=true

## SWT/OLE

OleControlSite, OleAutomation提供一套支持XActive的SWT控件，可显示Flash

但只支持Win32，Linux下的SWT包中没有OLE。

## IE显示flash