OpenJUMP Plug-In Programmer's Guide The JUMP Pilot Project

Table Of Contents

Introduction	4
How To Become An OpenJUMP Plug-In Programmer	4
Five Steps To Getting Set-Up As A Plug-In Programmer	5
What You Need To Know	5
Opportunities For Extension	5
Core Development Versus Plug-Ins	
Part 1: OpenJUMP Plug-In Programming	
What Is A Plug-In?	
The PlugIn Interface	
The AbstractPlugIn Class	
Accessing OpenJUMP Internal's From A Plug-In	
The PlugInContext Class	
The WorkbenchContext Class	
Using Blackboards	
Memory Usage and Plug-In Initialization	
Packaging And Distributing Plug-Ins For OpenJUMP	
Creating an Extension For Your Plug-Ins	
Internationalization	
Distributing Your Plug-In	
Part 2: OpenJUMP DataSource Programming	
The DataSource Class	
The Connection Class	
The Old DataStore Class and Related Code	
Making Your DataSource Accessible Through the GUI	
Part 3: OpenJUMP CursorTool Programming	
The CursorTool Interface	
OpenJUMP Programmer's Guide	

How To Become An OpenJUMP Programmer Page 3 of 6

The AbstractCursorTool Class

Steps To Implementing A CursorTool

Part 4: OpenJUMP Renderer Programming

The Rendering Pipeline In OpenJUMP

Using the Style and/or BasicStyle Class

Implementing the Renderer Interface

Introduction

This document is a guide for users of OpenJUMP that would like to begin programming plug-ins for OpenJUMP. Although it assumes you have a basic knowledge of the Java programming language, it does explain trickier programming concepts and assumes no previous knowledge of OpenJUMP's application programming interface. A good portion of the material in this guide is pulled from or build directly on information contained in the original JUMP Developer's Guide. Other material contained in this guide is based on the experiences of programmer's familiar with the creation of plug-ins for OpenJUMP and modification or maintenance of OpenJUMP's core. If you have questions about this document, or suggestions on improving it, please send an e-mail to sunburned.surveyor@gmail.com.

This document begins with a brief review of what it takes to get started programming OpenJUMP plug-ins and the opportunities plug-in programmers have for extending OpenJUMP. It then splits a detailed discussion of the four (4) main ways the plug-in programmer can add functionality to OpenJUMP in four (4) separate parts.

How To Become An OpenJUMP Plug-In Programmer

Five (5) Steps To Becoming An OpenJUMP Plug-In Programmer

You've decided that you would like to create a plug-in for OpenJUMP. How do you get started with plug-in programming for OpenJUMP? How do you become an OpenJUMP programmer?

There are 5 steps to getting started:

[1] Select and install a Java IDE or Java source code editor.

Most OpenJUMP programmers use Eclipse as their Java IDE for OpenJUMP. Eclipse is an open source program like OpenJUMP, and can be easily downloaded for free from the web. Another popular (and open source) IDE for OpenJUMP is Netbeans. GUI development for OpenJUMP plug-ins can be simplified trough the use of the Netbeans. If you don't want to deal with a full-blown IDE, you might try a Java source code editor like JEdit, which is also open source.

If you aren't familiar with these programs it would be best to do a little research before installing them. Questions about the choice of Java IDE our Java source code editor for OpenJUMP programming would be appropriate for the OpenJUMP programmer mailing list, but questions on the installation and use of these programs typically would not be appropriate.

Eclipse Website: http://www.eclipse.org/

Netbeans Website: http://www.netbeans.org/

JEdit Website: http://www.jedit.org/

OpenJUMP Programmer's Guide The JUMP Pilot Project 2008-06-12 00.00.10 How To Become An OpenJUMP Programmer Page 5 of 6

[2] Download the Javadoc for OpenJUMP or know how to access it online. (You may also want to make sure that you have access to the Javadoc for JTS, the geometry library used in OpenJUMP, and to the Javadoc from other programming libraries that will be used by your plug-in.)

OpenJUMP Javadoc Online: http://jump-pilot.sourceforge.net/javadoc/openjump_javadoc/

JTS Source Code Download: <u>http://sourceforge.net/project/showfiles.php?group_id=128875</u>

(Currently the JTS Javadoc is not available online and must be downloaded with the source code.)

[3] Subscribe to the programmer and user mailing lists for OpenJUMP.

JPP Developer/Programmer Mailing List: <u>https://lists.sourceforge.net/lists/listinfo/jump-pilot-devel</u>

[4] Be familiar with the SourceForge Project page for the JUMP Pilot Project, the organization that maintains OpenJUMP.

JPP SourceForge Project Page: http://sourceforge.net/projects/jump-pilot/

[5] Obtain a copy of the orginal JUMP Developer Guide and make sure you have access to the OpenJUMP Wiki. This documentation will assist you with the creation of your plug-in.

Original JUMP Developer Guide: <u>http://www.jump-project.org/inc/JUMP/assets/JUMP_Developers_Guide.pdf</u>

What You Need To Know

What do you need to know to get started with the creation of plug-ins for OpenJUMP?

A good knowledge of the Java programming language will help. The more Java programming you know, the easier it will be to create plug-ins for OpenJUMP. However, some very simple plug-ins can be prepared with only a basic knowledge of Java.

It is also helpful to be familiar with basic principles of working with geospatial data. This includes how real world objects can be represented as simple features and the basics of vector geometry.

Opportunities For Extension

OpenJUMP has been designed in a way that allows for many opportunities to extend the functionality of the program without the need to "build" the entire program from source code. This allows the plug-in programmer to accomplish a great deal without modifying OpenJUMP's internals.

OpenJUMP Programmer's Guide The JUMP Pilot Project 2008-06-12 00.00.10 There are currently four main ways that OpenJUMP can be extended by a plug-in programmer.

- [1] By implementing a new plug-in.
- [2] By implementing a new renderer.
- [3] By implementing a new cursor tool.
- [4] By implementing a new datasource.