Java TreeView Programmer's Guide

Alok Saldanha

Java TreeView Programmer's Guide

Alok Saldanha

Table of Contents

Preface

This manual is designed to get you started developing Java TreeView derivatives. This consists of two things:

1. Getting the tools necessary to develop Java TreeView

2. An Overview of the archtecture

Detailed descriptions of classes and stuff can be found in the Javadoc, and probably change too quickly to be worth documenting in a separate manual.

An outdated description of the architecture can be found in my thesis, which is available online. Until we get real funding, I will focus on putting useful reference material in here. This information will not be useful to you unless you are willing to read a lot of source code.

Chapter 1. Getting Started

This chapter will get you started with developing java treeview.

Tools required for developing Java TreeView

- Java SDK
- JEdit or other code editor
- ant, a Java-based replacement for make
- DocBook environment, *only for documentation authoring*

Required Tools

In this section, I will describe the tools required to build java treeview.

TreeView Source Code

You must, of course, download the source distribution of Java Treeview from http://jtreeview.sourceforge.net

Java SDK

The java sdk can be downloaded from the sun website at http://java.sun.com/j2se/.

ant

Apache Ant, the build tool, is available from http://ant.apache.org/bindownload.cgi

Code Editor

The crossplatform editor jedit is recommended, available from http://www.jedit.org/index.php?page=download

DocBook

Docbook is more of a file format (an XML dialect to be precise) than a tool.

Building Java Treeview

Once you have the java sdk and apache ant installed, unpack the source distribution of java treeview, open a command line shell, switch to the unpacked source directory, which should contain the build.xml file, and type "ant dist" to build java treeview. You should now have a freshly built TreeView.jar sitting in the "dist" subdirectory.

Building the documentation

Download a recent version of docbook-xsl or docbook-xsl-ns from http://sourceforge.net/project/showfiles.php?group_id=21935 [http://sourceforge.net/project/showfiles.php?group_id=21935] unpack,

and set an environment variable $XSL equal to the path. In mac os x, if docbook-xsl is unpacked in /usr/local/share/xml/xsl/docbook-xsl, then you would set the path with **export XSL=/usr/local/share/xml/xsl/docbook-xsl**.

Download the current version of docbook (or docbook-4.5.zip, which is known to work) from http://www.oasis-open.org/docbook/, unpack and make a link called "docbook" in the LinkedView directory. This will enable your xml source files to be checked against the DTD, which greatly helps with writing the source when you have a DTD aware editor (such as XML Buddy [http://xmlbuddy.com/] for Eclipse). This is also required to resolve external entities, such as "é".

You will probably also need the ISO entity sets from http://www.oasis-open.org/docbook/xmlcharent/0.3/index.shtml. Put them in an "ent" subdirectory of the docbook directory (i.e. the docbook-4.5 directory, not the xsl direcgtory).

If you want to make the PDFs, you will need FOP (http://xmlgraphics.apache.org/fop/. After you unpack this file, set an environment variable $FOP to point at the directory containing the fop shell script, for example **export FOP=/usr/local/share/xml/fop**.

Make sure you have xsltproc installed. You should now be able to run the createAllDoc.pl script.

Chapter 2. Reference Material

This chapter will hold reference material for java treeview development

Structure of XML Configuration Files

There are two types of xml configuration files used by java treeview, a global xml configuration file and a per-document configuration file or JTV. The location of the global xml configuration file is described in the user manual.

Structure of Global XML Configuration File

There is a single global XML configuration file in which java treeview stores the following information

1. recently used files list
2. last style used
3. whether parse quoted strings was used
4. defaults settings for views

Java Treeview 1.1.1 Global XML config

- ProgramConfig

  Root node of XML config

  - FileMru arributes (style, quotes)

    Node holds recent files, style attribute indicates last style used to open file, quotes indicate whether parseQuotedStrings was selected

    - File attributes (root, dir, style) optional (cdt)

      Node represents an individual file, with the dir it is found in, the root of the filename, and the style to open with. If cdt is specified, it means to use that extension instead of cdt to find the GCDT file.

  - GeneUrlPresets attribute (default)

    Holds default gene url presets, default attribute indicated default preset. -1 means do not link at all by default.

    - Preset attributes (name, template, header, enabled)

      Holds info for one url preset. Name is the name of the preset, template is the template to fill in, header is which header to fill it in with (by name, not index) and enabled is whether the preset is enabled. I'm not sure when or why I added an enabled flag to all presets.

  - ArrayUrlPresets attribute (default)

    Holds default array url presets, default attribute indicated default preset. -1 means do not link at all by default.

    - Preset attributes (name, template, header, enabled)

Holds info for one url preset. Name is the name of the preset, template is the template to fill in, header is which header to fill it in with (by name, not index) and enabled is whether the preset is enabled. I'm not sure when or why I added an enabled flag to all presets.

- Plugins

  Presets and defaults for particular plugins.

  - PluginPresets attrbutes (name)

    presets for plugin identified by name. Nodes will be provided to "presets configuration" dialogs that plugins make available, as well as the plugins themselves.

  - PluginDefaults attributes (name)

    Defaults for particular plugin, identified by name. These defaults shadow the configuration of the plugin in the per-document jtv, and are not directly modified by the plugin.

- Registration

  registration status

  - Entry attributes (jtv_version, java_version, java_vendor, os_name, os_arch, os_version, install_ip, install_host, install_date, status, first_name, last_name, email, institution, contactOkay)

    Registration entry, with various info about the installation. status keeps track of the status of that registration, and has values "deferred", "declined", "pending" and "complete". The meaning of these settings, as of all settings, is currently defined only in the source code of java treeview.

Java Treeview 1.0.12 Global XML config

In Java Treeview up until 1.0.12, the global configuration did not make any real distinction between places in which view-specific default values were stored, and places where program-wide defaults were stored. Moving forward, view specific defaults are now placed in a special "Defaults" node, which has subnodes for each type of view, and view-specific presets are put in a special "Presets" node. There are two things that are retained as view-indpendent general presets, the url and gene linking configuration. These pieces are actually provided to the views by the ViewFrame through the getUrlExtractor and getArrayUrlExtractor methods. Moving the per-view presets to their own nodes removes management of the dialogs from the main program, as well as weird calls such as "getKaryoColorPresets" that are clearly used by only one type of view from the LinkedViewFrame interface.

The following is the structure of the global configuration as of JTV 1.0.12

- ProgramConfig

  Root node of XML config

  - FileMru arributes (style, quotes)

    Node holds recent files, style attribute indicates last style used to open file, quotes indicate whether parseQuotedStrings was selected

    - File attributes (root, dir, style) optional (cdt)

      Node represents an individual file, with the dir it is found in, the root of the filename, and the style to open with. If cdt is specified, it means to use that extension instead of cdt to find the GCDT file.

- GeneUrlPresets attribute (default)

  Holds default gene url presets, default attribute indicated default preset. -1 means do not link at all by default.

  - Preset attributes (name, template, header, enabled)

    Holds info for one url preset. Name is the name of the preset, template is the template to fill in, header is which header to fill it in with (by name, not index) and enabled is whether the preset is enabled. I'm not sure when or why I added an enabled flag to all presets.

- ArrayUrlPresetsattribute (default)

  Holds default array url presets, default attribute indicated default preset. -1 means do not link at all by default.

  - Preset attributes (name, template, header, enabled)

    Holds info for one url preset. Name is the name of the preset, template is the template to fill in, header is which header to fill it in with (by name, not index) and enabled is whether the preset is enabled. I'm not sure when or why I added an enabled flag to all presets.

- ColorPresets

  color presets for dendrogram

  - ColorSet attrbutes (name, up, down)

    Set of colors to use for dendrogram view. name is name, up is up color, down is down color.

- KaryoColorPresets

  color presets forkaryoscope

  - KaryoColorSet attribute (name)

    Set of colors for karyoscope view. name is name of set.

    - Color attribute (type, hex)

      Particular color in set. Type is type of color, hex is hex value for color.

- ScatterColorPresets

  sctterview color presets

  - ScatterColorSet attribute (name)

    Set of colors for karyoscope view. name is name of set.

    - Color attribute (type, hex)

      Particular color in set. Type is type of color, hex is hex value for color.

- CoordsPresets

  coordinates presets for karyoscope

- Registration

  registration status

  - Entry attributes (jtv_version, java_version, java_vendor, os_name, os_arch, os_version, install_ip, install_host, install_date, status, first_name, last_name, email, institution, contactOkay)

    Registration entry, with various info about the installation. status keeps track of the status of that registration, and has values "deferred", "declined", "pending" and "complete". The meaning of these settings, as of all settings, is currently defined only in the source code of java treeview.

Structure of JTV XML Configuration File

Each document also has a local JTV file for document-level settings. The nodes i

- UrlExtractor attribute (urlTemplate,isEnabled, index)

  used for gene url linking

- ArrayUrlExtractor attribute (urlTemplate,isEnabled, index)

  used for array url linking

Architecture of Karyoscope Plugin

The main class of the Karyoscope Plugin is KaryoPanel. It is a container for the KaryoView, KaryoViewParameterPanel and a StatusPanel, and coordinates interaction between them.

Architecture of Dendrogram Plugin

Url Extraction

There is a single instance of the UrlExtractor for each document that is loaded by java treeview. Each instance contains a reference to the application level UrlPresets. Internally, there are two ways that a URL extractor can resolve a request to make a URL, with the specific URL for a given HEADER (via lookup of the column header in the presets) or via the generic URL preset.