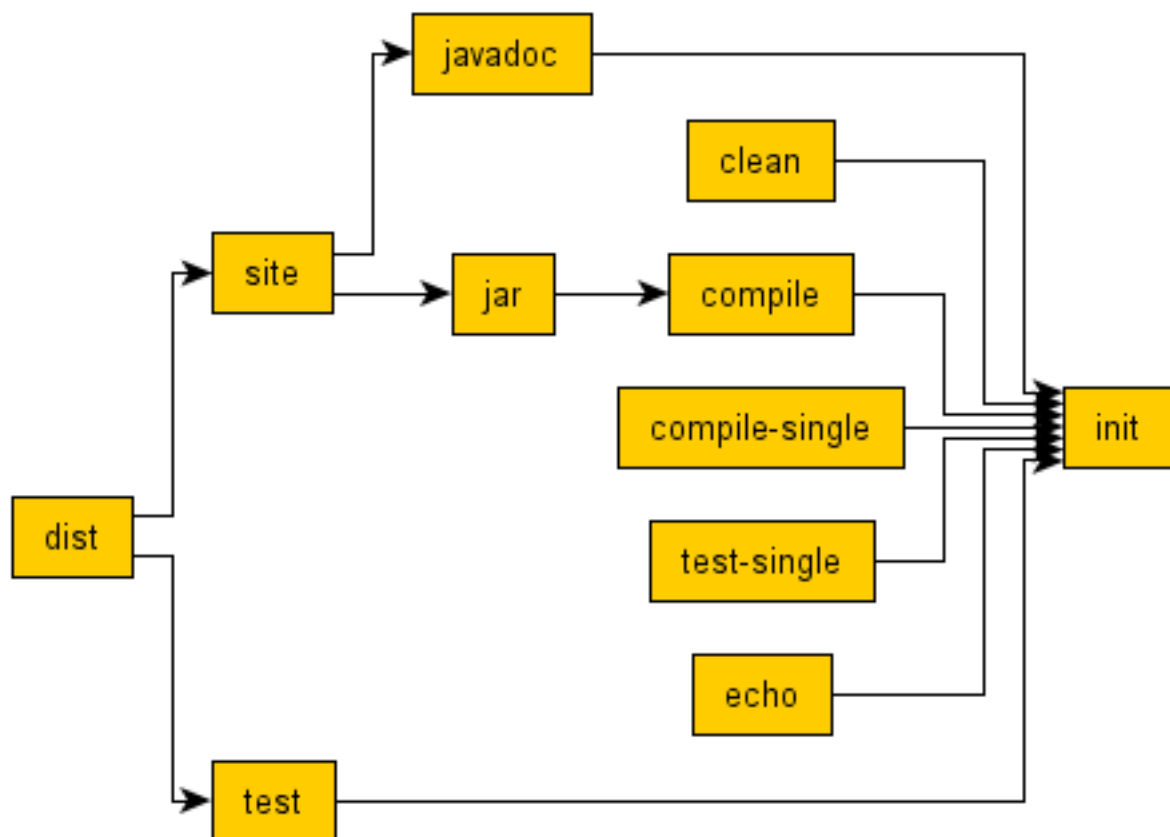


GraphTK 1.0 - Developer Manual

Julius E. Adorf



Graph generated with GraphTK.
<http://graphtk.sourceforge.net>

Visualized with yEd from yWorks.
<http://www.yworks.com/>

Table of Contents

About GraphTK

About this manual

Installation instructions

Graph formats

Generating graphs

External libraries

Known limitations

Support

About GraphTK

Introduction

GraphTK is a graph generation toolkit. Its main application areas are the analysis of build systems (e.g. Make or Ant), and the analysis of Java package dependencies.

The graphs may subsequently be visualized or processed with any external tool that understands one of the various graph formats supported by GraphTK.

GraphTK is an open source project hosted by SourceForge.net and published under the GNU General Public License. You may freely download and use this toolkit.

GraphTK may be used as a bridge between architectures that are based on graphs.

Target audience

The current version of GraphTK targets developers who are getting tired of doing dependency analyses that could be performed automatically.

Programming language

The toolkit is written in pure Java making it portable and platform-independent. Java developers may also use this toolkit as a library within their software packages.

About this manual

This manual describes the GraphTK software package. It is intended to be a reference for developers. It neither replaces the Javadoc nor covers all features but gives some examples and shows how to solve common problems.

This developer manual shows how to use the library within your Java code. If you are looking for a more general reference, please read the user manual.

Installation instructions

Requirements

GraphTK requires Java 5.0 to run. If you do not have installed Java JDK 5.0 yet, please download it from <http://java.sun.com/j2se/1.5.0/download.jsp>. If you are an end user, the Java Runtime Environment (JRE) is sufficient.

Steps

Simply extract the contents of the compressed ZIP file to the desired location, and you are ready to go.

Graph formats

GraphTK supports several formats for representing graphs: Ant, GML, XGML, GraphML and GraphML for yEd. If you need another format, please post a feature request. See section "Support" for details.

The format identifier is a name that is used throughout GraphTK to identify the graph format.

Note that not every format is equally suitable for using it with other applications.

Installing a new format

```
// Create a new definition providing a name, an identifier and a writer implementation
GraphFormatDefinition myGraphFormat = new GraphFormatDefinition("MyGraphFormatName",
                                                                "mygraph", new MyGraphFormatGraphWriter());

// Now register the new format.
GraphFormatDefinition.define(myGraphFormat);

// ...

// You can retrieve the definition in the following way.
try {
    GraphFormatDefinition.getGraphFormatDefinition("mygraph");
} catch (IllegalArgumentException iae) {
    System.err.println("Cannot find format definition for MyGraphFormatName");
}
```

Generating graphs

The following code snippet illustrates how to generate an Ant dependency graph:

```
try {
    // Create new generator
    BuildSystemGraphGenerator generator = new BuildSystemGraphGenerator();

    AntReader antReader = new AntReader();
    File antFile = new File("build.xml");
    BuildSystem antProject = antReader.read(antFile.toURL());
    generator.setBuildSystem(antProject);

    generator.setGraphMode(GraphMode.DEPENDENCY);
}
```

```

generator.setFilter(false);

// Generate graph
DirectedGraph g = generator.generateGraph();

// Serialize graph
OutputStream out = new BufferedOutputStream(new FileOutputStream(
    "build-graph.gml"));
GmlGraphWriter writer = new GmlGraphWriter(out);
writer.write(g);
} catch (GraphGenerationException gge) {
    System.err.println("Cannot generate graph: " + gge);
    gge.printStackTrace();
} catch (IOException ioe) {
    System.err.println("Cannot write graph: " + ioe);
    ioe.printStackTrace();
}
}

```

External libraries

Ant

Apache Ant is a Java-based build tool.

<http://jakarta.apache.org/commons/cli/>.

CLI

CLI is a useful library when writing a command-line interface.

<http://jakarta.apache.org/commons/cli/>.

JGraphT

JGraphT is used for the modelling of the graphs.

<http://jgrapht.sourceforge.net/>.

Swing Layout

Swing Layout is a useful library when writing a graphical user interface.

<https://swing-layout.dev.java.net/>.

Known limitations

Makefile parsing

The included makefile parser does not support all features of a makefile.

Make parser

- It does not understand macros.

There may also be some other limitations, which are not known yet.

Java package graph

The implementation of the source code scanner is quite straight-forward but not correct. The following conditions must be fulfilled to guarantee the generator works properly:

- import statements must be on separate lines
- import statements must not have comments between keyword, statement and finishing semi-colon
- package names must start with a lower case character
- class names must start with a capital letter

Static imports (Java 1.5.0) are ignored. Future versions will hopefully fix those issues.

Support

If you are running into problems, please have a look at the FAQ list on our web-site. If this does not solve your problem, please post a support request at our SF website, or send an e-mail to [graphtk\(AT\)municware.de](mailto:graphtk(AT)municware.de).

If you find a bug or like to have a specific feature implemented, feel encouraged to contact us and help us to improve this toolkit.

Links

Project home page: <http://graphtk.sourceforge.net/>

Project SF home page: <http://www.sourceforge.net/projects/graphtk>

Bug reporting: http://sourceforge.net/tracker/?group_id=161303&atid=819295

Support requests: http://sourceforge.net/tracker/?group_id=161303&atid=819296

Feature requests: http://sourceforge.net/tracker/?group_id=161303&atid=819298