JavaTools

User Manual and Reference Guide

Introduction

Features at a Glance

The JavaTools package provides a set of *Mathematica* functions that allow you to:

- Write/Paste full-featured Java code as *Mathematica* strings, and in one step compile and set up a symbol to call the Java code – without the need the load external libraries.
- Same with C# and F# code compilation on appropriate Windows/.Net systems.
- Use advanced Java implementations of multi-threaded American Option pricing methods with discrete dividends.
- Call advanced Java implementations of network flow / combinatorial optimization functions for thousands of nodes (problems so big that *Mathematica* alone cannot handle them).
- Harness the Google Collections (part of Google Guava) with *Mathematica*-specific implementations.
- Use Java Swing tables for viewing and editing of tabular data.
- Call various convenience functions.
- study *Mathematica* symbols with the symbol browser and expression browser.

About

JavaTools
Version 2.0

http://www.lauschkeconsulting.net

Table of Contents

Installation

Requirements

- JavaTools 1.0 or later. Available from www.lauschkeconsulting.net.
  - Current version is JavaTools 2.0.
- Google Guava. The JavaTools distribution contains the latest GoogleGuava library, and it is also available for download from http://code.google.com/p/guava-libraries/
  - Current version is Guava 14.0.1.
- *Mathematica* 6 or later. Available from www.wolfram.com.
  - Current version is *Mathematica* 9.
- Java 7 or later. Available from www.java.com.
  - Current version is Java 7 update 17. Current security baseline is Java 7 update 17.
  - If you want Java code compilation from JavaTools, you need the JDK, not just the JRE. The JRE does not have a compiler. You can use the JRE as well, but then you will not have Java code compilation.
- If you are using Windows, you need .Net 4. If you want C# code compilation, you also need a C# compiler, not just .Net. If you want F# code compilation, you also need an F# compiler.
- If you want Scala code compilation, you need either a Scala installation or SBT (Simple Build Tool).

Mathematica 9 has Java 7 bundled with it. *Mathematica* 7 and *Mathematica* 8 have Java 6 bundled with it, *Mathematica* 6 has Java 5 bundled with it. In either case, you have to use the JavaTools reinstall option or set the WRI_JAVA_HOME environment variable (see below) to ensure that JavaTools uses the latest Java. As the current security baseline is Java 7 update 13, JavaTools supports none of the Java version that ship with *Mathematica*.

Installation and Configuration

- Install/verify you are using Java 7 update 13. From the command line, submit

  ```
  java -version
  ```

  It should say something like

  ```
  java version "1.7.0_17"
  Java(TM) SE Runtime Environment (build 1.7.0_17-b02)
  Java HotSpot(TM) 64-Bit Server VM (build 23.7-b01, mixed mode)
  ```

  Note the 1.7.0_17. That means Java 7 update 17.

  If you want Java code compilation from JavaTools, you need the JDK (Java Development Kit), not just the JRE (Java Runtime Engine). The JRE does not have a compiler. You can use the JRE as well, but then you will not have Java code compilation as there is no Java compiler.

- Create a directory called JavaTools under your user home directory. This is $HomeDirectory, not $UserBaseDirectory. Place all files from the JavaTools distribution in the JavaTools directory. In other words, all files from the JavaTools distribution should go into

  ```
  FileNameJoin[{$HomeDirectory,"JavaTools"}]
  ```

- In the file javatoolsconfig.m set the location of your Java executable in the InstallJava[] and ReinstallJava[] lines. Note that you have to use the full path to the Java executable.

  On Linux it would look something like this:

  ```
  SetOptions[ReinstallJava,
  CommandLine->"/usr/java/latest/bin/java",JVMArguments->"-Xms4096m -Xmx4096m"];
  SetOptions[InstallJava,
  CommandLine->"/usr/java/latest/bin/java",JVMArguments->"-Xms4096m -Xmx4096m"];
  ```

  On Windows it would look something like this:

  ```
  SetOptions[ReinstallJava,
  CommandLine->"C:\\Program Files\\Java\\1.7.0_17\\bin\\javaw.exe",
  JVMArguments->"-Xms4096m -Xmx4096m"];
  SetOptions[InstallJava,
  CommandLine->"C:\\Program Files\\Java\\1.7.0_17\\bin\\javaw.exe",
  JVMArguments->"-Xms4096m -Xmx4096m"];
  ```

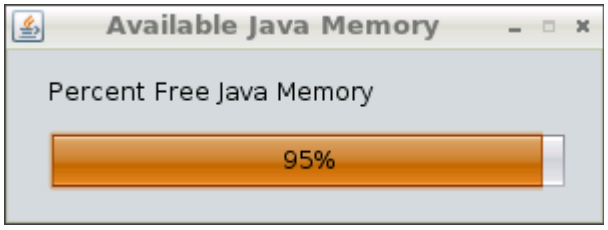
Use whatever values suit you for the memory configuration, or leave out to use default values. Xms is used for the start-up memory, and Xmx is used for the maximum memory allocated to Java.

Specify any additional runtime configuration options you want to use (memory, debugging, aggressive method compilation, etc.) in the JVMArguments rule as well.

- If you do not want to use the memory monitor, set the variable

  usememorymonitor

  in the file javatoolsconfig.m to False, or remove the line. (Default is False)



- On Windows and mono systems, to enable .Net 4.0 for *Mathematica*, find the file InstallableNET.exe in

  FileNameJoin[{$InstallationDirectory, "SystemFiles", "Links", "NETLink"}]

  and add a new line

  <supportedRuntime version = "v4.0"/>

  in the startup section above all other lines already present. Versions are tried in the order that they appear in the file, so place the preferred version first. The file will then look something like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
 <startup>
  <!-- The supportedRuntime lines control which version of the .NET Framework will
     be used when .NET/Link is launched with InstallNET[].
     .NET/Link requires at least .NET 2.0. If you have .NET 3.0 installed,
     it will be used (note that the 3.0 version is really just the 2.0
     version with some extra assemblies).
    -->
   <supportedRuntime version="v4.0" />
   <supportedRuntime version="v2.0.50727" />
 </startup>
</configuration>
```

You can check your .Net version, as recognized by *Mathematica*, with:

```
Needs["NETLink`"];
InstallNet[];
LoadNETType["System.Environment"];
System`Environment`Version@ToString[]
```

or

```
Needs["NETLink`"];
ShowNETConsole[]
```

- Note that on .Net or mono systems you need a C# compiler installed for the C# compilation from JavaTools to work, and you need an F# compiler installed for the F# compilation from JavaTools to work.

- If you want Scala code compilation from JavaTools, you need either a Scala installation (recommended), or you can use the Simple Build Tools (SBT):

  http://www.scala-sbt.org/

  Follow the instructions in the JavaTools config file (that is javatoolsconfig.m, see above) to use a full Scala installation or SBT. With a full Scala installation, you need to set one variable, with SBT you need to set three.

  Installation Instructions for SBT:
  https://github.com/harrah/xsbt/wiki/Getting-Started-Setup

Look-and-Feel Configuration

JavaTools supports all standard and most third-party add-on/plug-in look-and-feels. To install a third-party look-and-feel, download the .jar file from the third-party supplier, place it in the JavaTools directory, identify the name of the entry point class, and place a line

laf="<full name space class name to entry point class>"

in a text file lookandfeel.m in the JavaTools directory. Example:

laf="com.jgoodies.looks.plastic.Plastic3DLookAndFeel"

This will enable the look-and-feel immediately for the *Mathematica* package. To also use the look-and-feel from Calc (through the OpenOffice plug-in/extension), add the .jar file to the OpenOffice class path, as described in steps 3 and 4 in the previous section.

If no file lookandfeel.m exists in the JavaTools directory or laf is assigned the string "default", JavaTools will use the system's default look-and-feel.

For a good overview of various free and commercial look-and-feels, visit
http://www.javootoo.com

The following Look-and-Feels have been tested to work with JavaTools:

- JGoodies Plastic3D
- JGoodies PlasticXP
- JGoodies Plastic
- JGoodies Windows
- Office2003 (Windows only)
- OfficeXP (Windows only)
- VisualStudio2005 (Windows only)
- Nimrod
- Fh
- Tiny
- Tonic
- Tonic Slim
- Infonode
- Napkin
- SquareNess
- EaSynth

which can be downloaded from javootoo.

The Alloy look-and-feel has also been tested to work with JavaTools, which can be obtained from http://lookandfeel.incors.com/.

Getting Started

Loading the Package

To start using JavaTools, you must first load the JavaTools package.

With *Mathematica* version 6 and above:

```
Get@ToFileName[{$HomeDirectory, "JavaTools"}, "JavaTools .m"]
```

With *Mathematica* version 7 and above:

```
Get@FileNameJoin[{$HomeDirectory, "JavaTools", "JavaTools .m"}]
```

Features

The JavaTools Features are explained on the website.

Java, Scala, C#, and F# Code Compilation
http://www.lauschkeconsulting.net/jcc.html

American Options
http://www.lauschkeconsulting.net/americanoptions.html

Google Collections
http://www.lauschkeconsulting.net/jt.html

Network Flow / Combinatorial Optimization Functions
http://www.lauschkeconsulting.net/networkflow.html

Math Functions
http://www.lauschkeconsulting.net/math.html

Java Swing Table Viewer and Editor
http://www.lauschkeconsulting.net/jtableview.html

Java Windows
http://www.lauschkeconsulting.net/javawindows.html

Convenience Functions / Utilities
http://www.lauschkeconsulting.net/utility.html

Symbol Browser and Expression Browser
http://www.lauschkeconsulting.net/symbolbrowser.html
http://www.lauschkeconsulting.net/expressionbrowser.html