YATT

Yet Another Tracer Tool

by Dieter Dirkes, Wily Technology

Documentation

Date: 2005.01.11

Contents

1 Introduction

1.1 Why YATT

I scanned and reviewed all field tools available to create PBD files from our customer code. The result was not satisfying and I had lots of ideas of what I expect from a tracer creation tool.

1.2 The name

Originally I named the stuff XTracerTool (for extended TracerTool) but then colleagues asked “why are you creating yet another tool”, so I named it YAT, then we came to “yet another tracer tool” YATT…

1.3 The technology behind

YATT is programmed with the usage of Eclipse. Due to optical limitations and the bad performance of Swing I dediced to use YATT as a chance for me to learn of how to program with SWT (Standard Widgets Toolkit), which was created for Eclipse by IBM originally. SWT uses the native GUI Widgets of the underlying operating system and therefore SWT GUIs are fast and they look like a real app from that OS.

2 Installation

My wish was to get an application that is easy to use, so the idea of using an executable JAR file was near. Unfortunately Java does not support internal structures of executable JAR archives, you can not have a /lib directory in a JAR file and all libraries in this directory are automatically used. So I had to create a small own boot mechanism with an own classloader who solves the dependencies from this internal libraries.
The result is only one JAR archive that contains everything needed to run YATT.
Just create a directory for YATT and put the JAR into this directory. YATT normally creates a tmp directory where the temporary unpacked libs and the parsed sources are stored.

3 Running YATT

To start YATT, just use a 1.4.X JVM and run:

```
java -jar yatt.jar
```

The current version only runs under Windows OS and Linux. Please let me know if you have any problems running YATT on these platforms.

Caution
YATT uses a temporary directory to unpack and use the internal libraries. This directory is named ./tmp in the current directory where you run YATT per default.

Logging
When YATT detects that it runs with its own BootStrap it redirects System.out and System.err to an Log4J logfile named error.log.

4 Using YATT

4.1 Concepts

YATT loads the stuff that you deploy to your JVM or AppServer: Any archive containing compiled classes, like a EAR, JAR, WAR or ZIP file. The tool loads all classes from this archive and from all archives that are contained in the original one.

All work in YATT is session based. So you can load several compiled sources into a session and work on these.

4.2 Opening compiles sources

By selection File -> Load Compiled Sources you load the sources into YATT. While loading YATT is resoluting the hierarchical relations between the classes and interfaces loaded.
The result is shown in the class tree immediately.

4.3 Sessions

All work in YATT is session based. So you can load several compiled sources into a session and work on these. By selecting File -> New Session you clear the old session and create a new one.

4.4 PBDs & PBLs

The internal structures of YATT are already designed to support PBDs and PBLs. Within PBDs there is support for sections, so that you can apply your traces to different sections and as a result you get a well formatted and sectioned PBD file.
{These functionality is still in development}

5 Progress of Implementation // History

2004.08.05	now the class-tree shows also when a class is inheriting traces from its parents.
2004.08.05	fixed a bug. When you enable the "show interfaces" now, you have to klick on find again, as the visible tree has changed. I am not running the search automatically, as this may take time.
2004.08.05	enhanced the functions delivered by the internal ImageStore, so the modules do not have to deal with images directly.
2004.08.05	introduced filters.xml to define filters for skipping packages.
2004.08.09	finished implementation of filters and tested these with an Archive compiled for and with BEA WebLogic.
2004.08.11	fixed a bug (thanks to Rich Wilson). When you selected PBD-> Save As and pressed cancel you got a NullPointer exception. Also sPBD -> Save did not work right.
2004.08.11	fixed setting of currentSourceDirectory in config.xml and storing this value when opening a source with the GUI.openCompiledSource(). Now the currentCompiledSourceDirectory is saved in the config.xml.
2004.08.13	fixed display of Methods with Signatures. Also applying Tracers to Methods with Signatures did not work well. The Class-Outline now highlights the traced methods in the right way.
2004.08.26	Help Dialogs are now existing. The integration of the OS browser via SWT also works nice ☺
2004.08.26	BootLoader now also unpacks the yatt-manual.pdf to the libtempdir, so that the integrated browser can display this when you click on help…
2004.08.26	started implementation of PBDFileParser… IN PROGRESS
2004.09.06	implemented doublette filter for addTracers… now only one line per tracer and directive for each method / class is added.
2004.09.08	implemented selection of multiple source files in FileChooserDialog..
2004.09.29	Outline is now working with already created Images from ImageStore. So I can reuse the outline module for the Tree showing classes with methods.
2004.10.08	Changed view management by implementing a ViewManager.
2004.10.28	Added RecentFileList to yatt and fixed some bugs.
2004.11.04	Extended internal data structures (Directives,etc.)
2004.11.29	Modified the Directives.xml and the regarding reading & tracing mechanism.

2004.11.30	Reworked the booting mechanism. Now having a BootLoader.properties in the yatt.jar which defines some generic behavior and allowed me to clean up some bad mess in the booting mechanism.
2004.11.30	Extended the functionality of adding Groups to Classes. ContextMenu, Directive, DirectiveConfigReader, TracerGroupLevel + Reader, etc.
2004.12.04	Added TracerGroups.xml and its implementation.
2004.12.05	Fixed some minor problems.

6 Restrictions

6.1 Methods in Class Tree

When selecting “showMethods in ClassTree” the methods are shown only with a rudimental Method Image. For seeing details of a Method klick on the method or class and look at the right outline.

7 Wishlist

1. Overview of already loaded archives. Rejection of already loaded archives?
2. multiple select of JAR files... -> done on 08.09.2004

8 Credits

- My lovely wife Agata
- The Wily EMEA System Engineers and Managers
  - Simon Parker
  - Rich Wilson (rest in piece)
  - Andrea Ubbiali
  - Eric Serra
  - Rainer Schuppe
- Michael Teroerde from Accenture Frankfurt (Super Star ***)
- Dragan Zuvic from Debitel Stuttgart