Eclipse Memory Analyzer Release Review 0.8

Review Date: May 29, 2008

Community Channel:

- mailto:mat-dev@eclipse.org
- http://www.eclipse.org/newsportal/thread.php?group=eclipse.technology.memory-analyzer

Author: Andreas Buchen (project lead)

Introduction

- Memory Analyzer is a Technology sub-project in Incubation
  - http://www.eclipse.org/mat

- This release (part of the **Galileo Release Train**) adds support for IBM dumps (via DTFJ API), thread stack information, improved object inspectors and miscellenous bug fixes.

Features

- Report Memory Leak Suspects
- Calculate Retained Sizes
- Find who is keeping Objects Alive
- Group Objects to Detect Pattern
- Query Heap with an SQL-like Language
- Works with multi GB heap dumps
- Supports HPROF-formatted heap dump

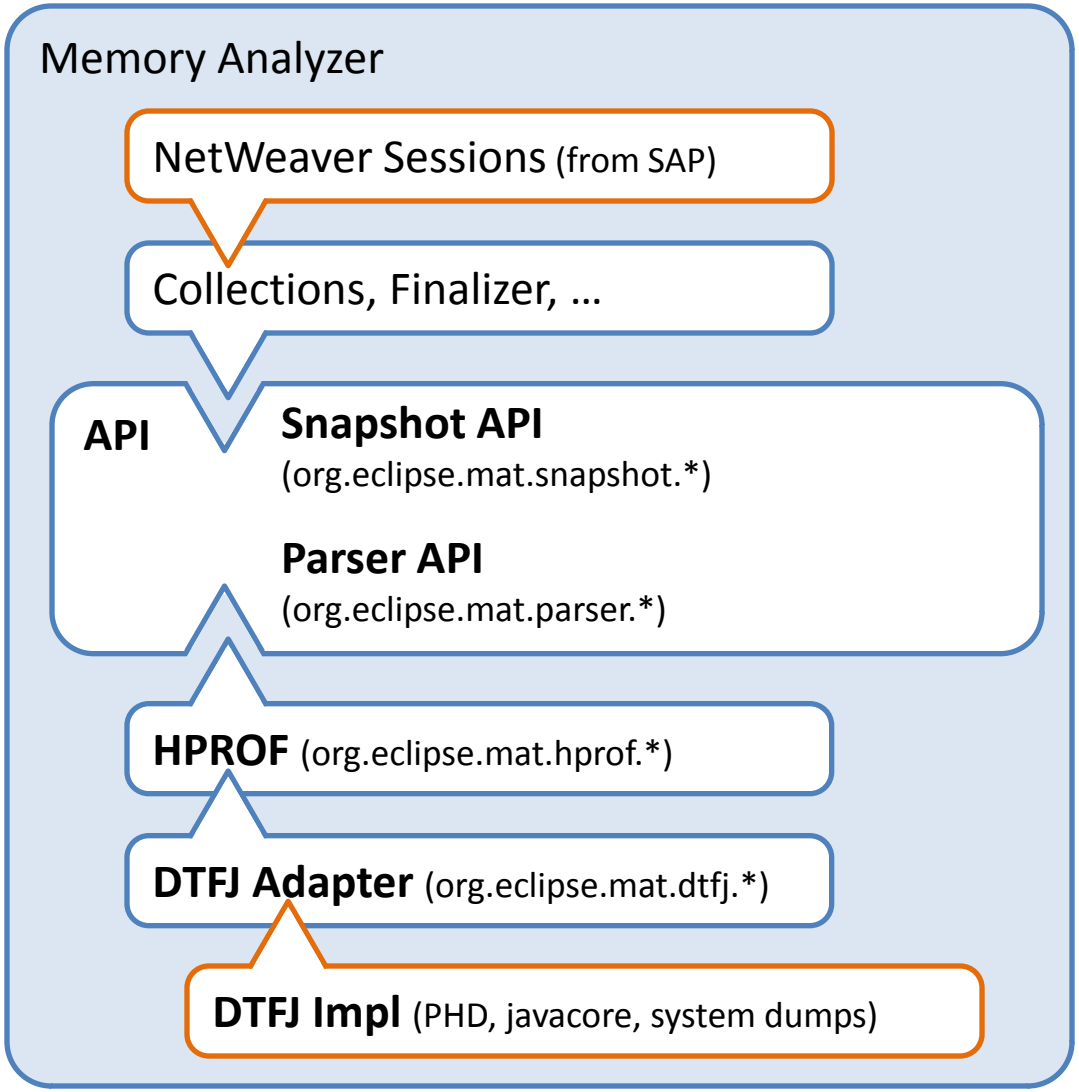
+ Supports IBM dumps via DTFJ (Diagnostic Tool Framework for Java)
+ Thread Stack Information + Java Locals
+ Improved Object Inspectors

Non-Code Aspects

- Documentation Is Generated Using DITA
  - provided via the help center
- Online Documentation via
  - WIKI http://wiki.eclipse.org/index.php/MemoryAnalyzer
  - Webinar http://live.eclipse.org/node/520
  - Blog http://dev.eclipse.org/blogs/memoryanalyzer
- Cheat Sheets

Summary: a Wealth of Material is is Available, but often brief and not easily accessible to non-domain experts.

APIs



The Memory Analyzer provides two major interfaces:

a) The **Snapshot API** provides access to the logical object graph inside the heap. It enables inspections that analyze collections, identify leak suspects etc.

b) The **Parser API** makes reading the raw heap dump format pluggable.

APIs conform with Eclipse Quality Standards.

MAT @ Eclipse.Org

(known) 3rd Party Extensions

Architectural Issues

Summary: Architecture is Settled and Performs Well on Multi-GB Heap Dumps

Tool Usability

Summary: Rich and Very Responsive UI. The Sheer Number of Heap Inspections can be Overwhelming for a Novice User.

End-Of-Life

This is the second release (still in incubation).
Nothing is end-of-live‘d.

Bugzilla

Bugzilla Usage Currently is Low, Features and Bugs Usually Reported via Newsgroup

18 bugs resolved

16 bugs open (enhancements + bugs)

Standards

MAT requires

- Execution Environment **J2SE-1.5**
- **Eclipse Platform 3.3** or greater
- **BIRT Chart Runtime 2.2.0** or greater

UI Usability

- Follow User Interface Guidelines
  - Multiple Language Support Now Done

Schedule

Release 0.8 – June 2009 – Galileo

Theme: Join Galileo Release, Support IBM dumps

Release 0.9 – Q4 2009

Theme: Comparing Heap Dumps, Usability, Documentation

Communities

- User Involvement Is Rare, but Forum Discussions and Feature Requests are Picking Up

- On Average, **800 Downloads Per Week** (Excluding Installations via Update Manager)
- **Newsgroup** (eclipse.technology.memory-analyzer) Shows Signs of Users Helping Each Other
- **Presentations** at Conferences
  - JAX (21 April 2009)
  - Eclipse Demo Camp Walldorf (19 May 2009)
  - JavaOne (3 June 2009)

IP Issues

- IP Process Followed
- IP Log
  http://www.eclipse.org/projects/ip_log.php?projectid=technology.mat
- Feature Contributions (besides bug fixes)
  - DTFJ adapter by Andrew Johnson, IBM
  - Re-create SWT images and colors from heap dump and show in Object Inspector
  - Externalize UI strings, label, messages for i18n
- Project Is Released Under EPL

Project Plan

Available at

http://www.eclipse.org/projects/project-plan.php?projectid=technology.mat

Themes

- Comparing Heap Dumps
- Usability Features
- Documentation Improvement
- Building A Community