

Eclipse Memory Analyzer Release Review

Review Date: Dec 10, 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 is the **First Release** as on Eclipse.Org
 - Provide Stable Version for External Extensions
 - Signal Maturity of the Project to New Users

Features

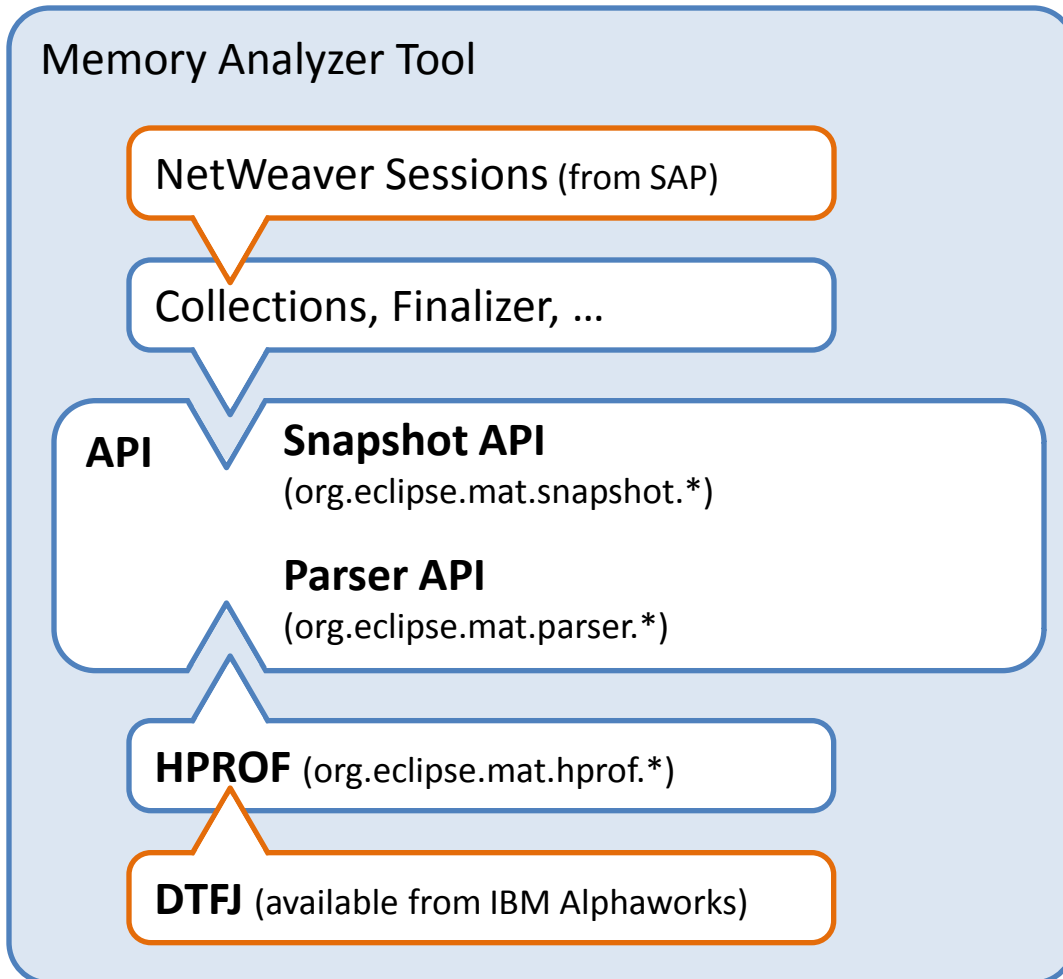
- Report Memory Leak Suspects
 - Big Objects, Big Threads
- Calculate Retained Sizes
 - Instantly for Objects, Fast for Arbitrary Object Sets, Optimized Approximation for Many Sets
- Find who is keeping Objects Alive
 - Path to/from Garbage Collection Roots
 - Immediate Dominators
- Group Objects to Detect Pattern
 - by Class Loader, by Arbitrary Attribute Values
- Query Heap with an SQL-like Language
- Works with multi GB heap dumps
- Pluggable Heap Formats, Heap Inspections, ...

Non-Code Aspects

- Documentation Is Generated Using DITA
- 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 Available, but often brief and not easily accessible to non-domain experts. This is one topic to improve within the next version.



APIs



The Memory Analyzer provides two major interfaces:

- The **Snapshot API** provides access to the logical object graph inside the heap. It enables inspections that analyze collections, identify leak suspects etc.
- 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 first release. Nothing is end-of-live'd.

Bugzilla

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

19 bugs resolved

8 bugs open (enhancements + bugs)

Standards

MAT requires

- Execution Environment **J2SE-1.5**
- **Eclipse Platform 3.3** or greater

UI Usability

- Follow User Interface Guidelines
- Exceptions:
 - Multiple Language Support Not Yet Done

Schedule

Release 0.7 – Nov 2008 – Initial Release

Theme: stable version

Release 0.8 – March 2009

*Theme: Comparing Heap Dumps, Usability,
Documentation*

Communities

- User Involvement Is Rare, but Forum Discussions and Feature Requests are Picking Up
- On Average, **450 Downloads Per Week** (Excluding Installations via Update Manager)
- **Newsgroup** (eclipse.technology.memory-analyzer) Shows Signs of Users Helping Each Other
- **Presentations** at Conferences
 - EclipseCon 2008
 - JavaOne 2008
 - Eclipse Summit Europe 2008 (Nov 20th)

IP Issues

- IP Process Followed
- IP Log
 - http://www.eclipse.org/projects/ip_log.php?projectid=technology.mat
 - No major feature contribution (yet) besides initial contribution
- 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