

Eclipse Memory Analyzer 1.2 Release Review

Community Channel:

- <mailto:mat-dev@eclipse.org>
- <http://www.eclipse.org/forums/eclipse.memory-analyzer>

Author: Krum Tsvetkov (project lead)

Introduction

- Memory Analyzer is a tool for heap dump analysis, which helps in analyzing memory leaks and high memory consumption of Java applications. It works fine with multi-million objects heap dumps and can quickly point to the suspicious objects and who is retaining them in the heap.
- Memory Analyzer was created as a sub-project under the Technology TLP in 2008 and moved under the Tools project in 2010
- It is part of the simultaneous release since Galileo
- <http://www.eclipse.org/mat>

Features

- Report memory leak suspects
 - Report memory waste –redundant Strings, empty collections
 - Calculate retained sizes
 - Find who is keeping objects alive
 - Query heap with an SQL-like language
 - Works with multi GB heap dumps
 - Supports various dump formats, e.g. HPROF, IBM PHD and system dumps
 - Thread stack Information + Java locals
 - Trigger heap dumps from within the tool
 - Compare any two or more table-formatted results
-
- + Support for `-XX:+UseCompressedOops` (precise object-size calculation)
 - + Syntax highlighting and code completion in OQL editor

Non-Code Aspects

- Eclipse Help Center documentation is generated using DITA
- Online documentation via
 - WIKI <http://wiki.eclipse.org/index.php/MemoryAnalyzer>
 - Webinars <http://live.eclipse.org/node/520>, <http://live.eclipse.org/node/939>
- Cheat sheets
- “Extending MAT Guide” (in Wiki):
 - http://wiki.eclipse.org/index.php?title=MemoryAnalyzer/Contributor_Reference#Writing_plugins_for_the_Memory_Analyzer

APIs

Memory Analyzer Tool

NetWeaver Sessions (from SAP)

Collections, Finalizer, ...

API

Snapshot API

(org.eclipse.mat.snapshot.*)

Parser API

(org.eclipse.mat.parser.*)


HPROF (org.eclipse.mat.hprof.*)

DTFJ (available from IBM Alphaworks)

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

- The Memory Analyzer tool is very helpful for troubleshooting of OutOfMemoryErrors. It can be also used proactively to analyze and reduce memory consumption.
- The tool provides rich and responsive UI.
- The sheer number of heap inspections can be overwhelming for a novice user.

End-Of-Life

- No features are removed
- No API has been deprecated / removed

Bugzilla

- Messages statistics
 - State 1.1, May 2011 (Total: 266, Open 59, Closed 207)
 - Current State, May 2012 (Total: 328, Open 71, Closed 257)
- Bugzilla used for discussions on new features / modifications

Standards

- MAT can work with:
 - HPROF binary dumps
 - IBM PHD dumps
 - IBM System dumps
- MAT Requires
 - Execution Environment **J2SE-1.5**
 - **Eclipse Platform 3.4 – 3.8, 4.2**
 - **BIRT Chart Runtime 2.3.0 or higher**

UI Usability

- Follow User Interface Guidelines
- Multi-language support
 - UI Strings are externalized via Eclipse NLS
 - Memory Analyzer is part of Babel
- Accessibility improvements made since 1.0
 - Tracked in https://bugs.eclipse.org/bugs/show_bug.cgi?id=300655
 - Followed guidelines: <http://www-03.ibm.com/able/guidelines/software/accesssoftware.html>

Schedule

- Release 1.2.0 - June 2012
 - Participate with 1.2.0 in the Juno release
- Release 1.2.1 - Juno SP1 (or earlier if urgent fixes are needed)
 - Bugfixes for 1.2.0
- Release 1.3.0 (or 2.0.0 if API needs to be changed) - Kepler, June 2013
 - Standalone MAT (RCP Application) based on Eclipse 4.x

Communities

- Contributors and committers
 - Committers (total number 5): 4 from SAP, 1 from IBM
 - *Active Committers are 2: 1 from SAP, 1 from IBM*
 - Recently increasing number of non-committer contributions done via Bugzilla
- Adopters
 - Integrated into SAP NetWeaverCE
 - Integrated into IBM Support Assistant
 - Integrated in Motorola's MOTODEV Studio for Android
 - Some additional extensions to MAT being written as shown by questions on the forum
- Users
 - A large user community (for the special niche of the tool)
 - The standalone RCP application has about 2800 downloads per week
 - Part of Indigo update site (downloads not counted)
 - Forums and Bugzilla used as communication channel

IP Issues

- All plugins contain appropriate license files
- All committers have completed Eclipse Committer Agreements and have been approved by the PMC
- All non-committer contributions are properly marked in Bugzilla
- IP Log is approved, attached to
 - https://bugs.eclipse.org/bugs/show_bug.cgi?id=380405

Project Plan

- Available at

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

Future Themes

- Provide standalone RCP MAT application based on Eclipse 4.x
- Research / provide comparison based analysis
- Detect more anti-patterns automatically