Chris Bailey, Java Support Architect

Debugging Java™ Applications

with Memory Analyzer and the IBM Extensions for Memory Analyzer

Overview

- IBM Monitoring and Diagnostics tool for Java™ – Memory Analyzer
- Dumps used by Memory Analyzer
- Analyzing problems with Memory Analyzer
- The IBM Extensions for Memory Analyzer
- Demo

IBM Monitoring and Diagnostics tool for Java – Memory Analyzer

- Part of the IBM Monitoring and Diagnostic Tools for Java™ collection of tools
  - A unified suite of tools to understand different aspects of Java applications
  - Available in the IBM Support Assistant
  - All fully IBM supported tools
- IBM version of the Eclipse Memory Analyzer Tool (MAT)
  - Modified to accept system dumps and IBM heap dumps
  - Further capabilities planned
- Tool that carries out memory analysis of Java applications
- Primary aim is memory leak detection and footprint analysis

Installing Memory Analyzer

- Installed into the IBM Support Assistant (ISA):
  - http://www-01.ibm.com/software/support/isa/
- Installed using the “Update → Find New... → Tools Add-ons” menu option
  - Under the “JVM-based Tools” section
  - Called “IBM Monitoring and Diagnostic Tools for Java – Memory Analyzer (Tech Preview)”
- Fully supported by the IBM Java Support team
  - Via normal IBM support routes

Dumps used by Memory Analyzer

- HotSpot based Java Runtimes (Solaris and HP-UX):
  - HPROF format heap dumps

- IBM Java Runtimes (AIX, Linux, Windows, z/OS):
  - System dump (core file, minidump, svcdump), processed by Jextract*
  - IBM PHD heap dump

Dump Format	Approx. Size on Disk	Objects, Classes, Class Loaders	Thread Details	Field Names	Field and Array References	Primitive Fields	Primitive Array Contents	Accurate GC Roots	Native Memory and Threads
IBM PHD	20% of Java heap size	✓	with Javacore*	✗	✓	✗	✗	✗	✗
HPROF	Java heap size	✓	✓	✓	✓	✓	✓	✓	✗
System dump	Java heap size + 30%	✓	✓	✓	✓	✓	✓	✓	✓

* by loading in a both a pair of files consisting of javacore.txt (IBM thread dump file) and a heapdump.phd that have been generated at the same time, thread details become available.

Acquiring Dumps: HotSpot Java Runtimes

- Interactive methods:
  - Using a Ctrl-Break:
    - When the -XX:+HeapDumpOnCtrlBreak command line option is set
    - HPROF format dump is generated along with a thread dump when a Ctrl-Break event is sent
  - Using the JMap tool:
    - jmap -dump:format=b <pid> (Java 5)
    - jmap -dump[live,]format=b,file=<filename> <pid> (Java 6)
  - Using the Operating System:
    - gcore command or the destructive kill -6 or kill -11 commands to produce a core file
    - Then, extract a heapdump from the core file using Jmap
    - jmap -dump:format=b,file=heap.hprof <path to java executable> <core>
  - Using the JConsole tool:
    - Under the HotSpotDiagnostic MBean in JConsole a dumpHeap operation is provided. This will request that a HPROF dump is generated.

- Event Based Methods
  - On an OutOfMemoryError:
    - When the -XX:+HeapDumpOnOutOfMemoryError command line option is set
    - no limit to how many heapdumps will be produced on this event per JVM run.

      Note: this has been changed in a recent update of HotSpot

Acquiring Dumps: IBM Java Runtimes

- Interactive Methods
  - Using a SIGQUIT or Ctrl-Break:
    - When a Ctrl-Break or SIGQUIT (usually generated using "kill -3") is sent to the IBM runtime
    - a "user" event is generated in the IBM dump engine
    - can however be configured to generate either a PHD format dump, or a System dump using the following command line options to the Java application
    - -Xdump:heap:events=user (PHD Heap Dump)
    - -Xdump:system:events=user (System dump)
  - Using the operating system to produce a system dump:
    - AIX: gencore (or the destructive kill -6 or kill -11)
    - Linux/Solaris: gcore (or the destructive kill -6 or kill -11)
    - Windows: userdump.exe
    - z/OS: SVCDUMP/Console Dump
  - Using IBM Monitoring and Diagnostics Tools for Java - Health Center:
    - The Health Center tool provides a menu option to request dumps from a running Java process
    - can request either a PHD or system dump

Acquiring Dumps: IBM Java Runtimes

- Event Based Methods
  - Using the IBM dump engine:
    - The dump engine provides a large number of events on which you can produce a "heap" (PHD) or "system" dump

Event	Description	Filtering	Example
gpf	General Protection Fault (crash)		-Xdump:system:events=gpf
user	User generated signal (SIGQUIT or Ctrl-Break)		-Xdump:system:events=user
vmstop	VM shutdown, including call to System.exit()	exit code	-Xdump:system:events=vmstop,filter=#0..#10
load	Class load	class name	-Xdump:system:events=load,filter=com/ibm/example/Example
unload	Class unload	class name	-Xdump:system:events=unload,filter=com/ibm/example/Example
throw	An exception being thrown	exception name	-Xdump:system:events=throw,filter=java/net/ConnectException
catch	An exception being caught	exception name	-Xdump:system:events=catch,filter=java/net/ConnectException
systhrow	A Java exception is about to be thrown by the JVM	exception name	-Xdump:system:events=systhrow,filter=java/lang/OutOfMemoryError, range=1..4
allocation	A Java object is allocated	size of object	-Xdump:system:events=allocate,filter=#5m

    - Exceptions can also be filtered on throwing method using '#'

      -Xdump:system:events=throw,filter=java/lang/NullPointerException#com/ibm/example/Example.bad

Acquiring Dumps: IBM Java Runtimes

- Event based methods
  - Using the IBM trace engine:
    - the trace engine allows PHD and system dumps to be triggered on method entry or exit
    - achieved using the following command line option to the application:
    - produces a system dump when the Example.trigger() method is called

      -Xtrace:maximal=mt,trigger=method{com/ibm/example/Example.trigger,sysdump}
    - produces a PHD dump when the Example.trigger() method is called

      -Xtrace:maximal=mt,trigger=method{com/ibm/example/Example.trigger,heapdump}
    - set a range so that you don't create dumps every time the method is called

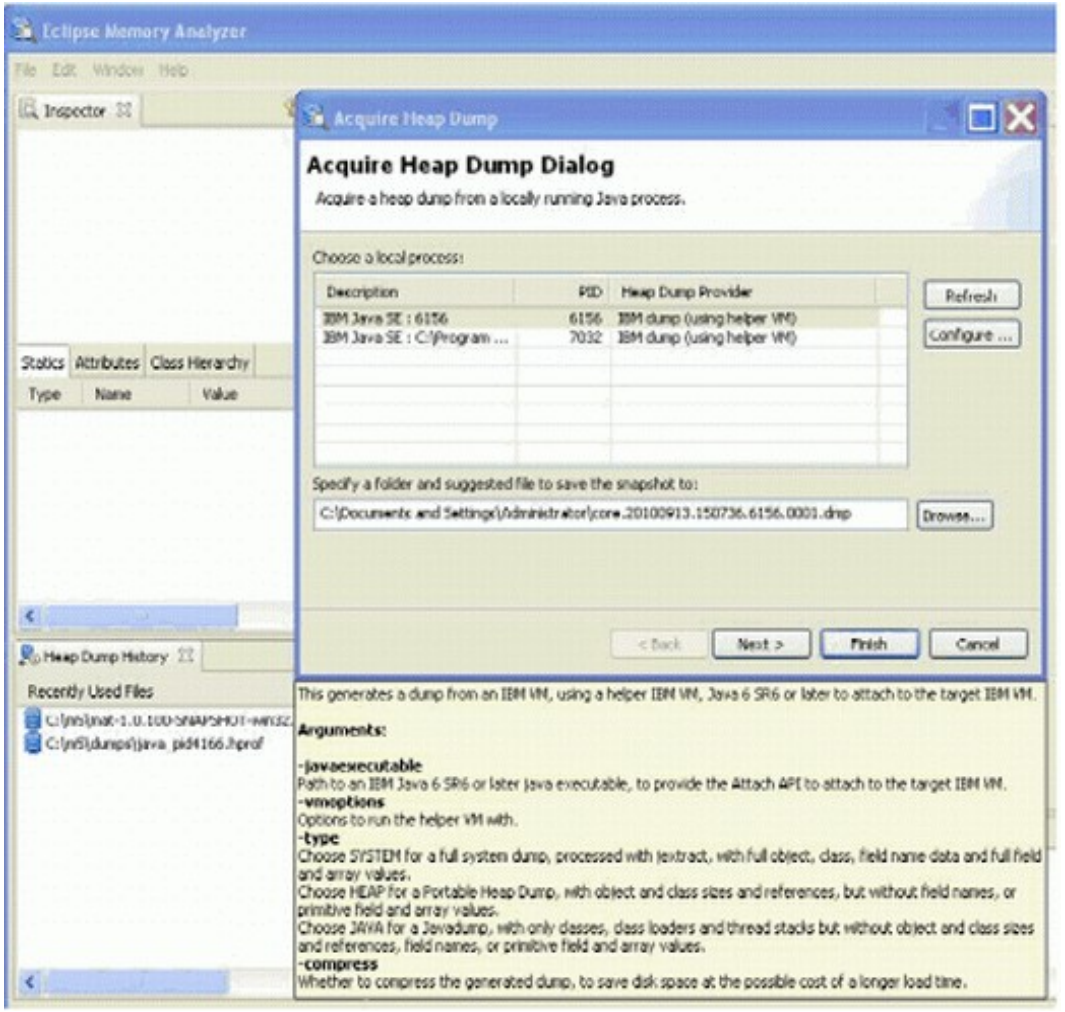
      -Xtrace:maximal=mt,trigger=method{com/ibm/example/Example.trigger,sysdump,,5,1}

- Programmatic Methods:
    - provide a com.ibm.jvm.Dump class with three methods:

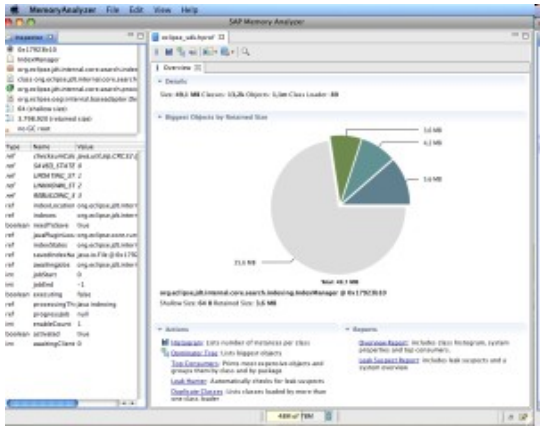
      javaDump(), heapDump() and systemDump()

Acquiring Dumps: Using Memory Analyzer

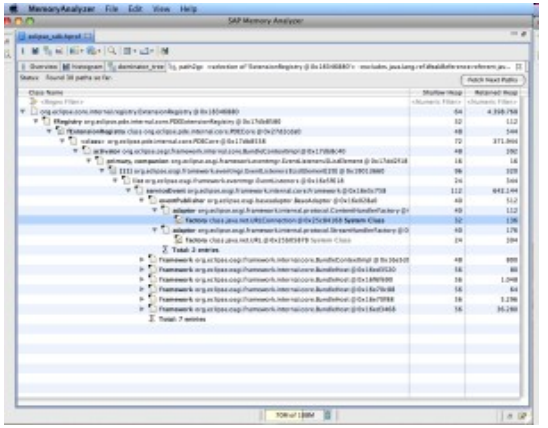
- Memory Analyzer also provides an "Acquire Heap Dump" option that allows you to trigger and load a snapshot dump from a Java process running on the same machine as Memory Analyzer.
- On HotSpot based runtimes the Memory Analyzer generates the dump using Jmap.
- For the IBM runtimes the dump is generated using the "late attach" functionality and the the programmatic API.



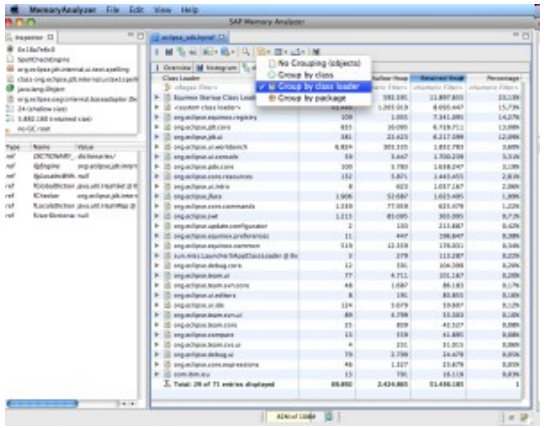
Analyzing problems with Memory Analyzer



Overview:
Overview of the heapdump including size and total number of objects.
Provides links to continued analysis



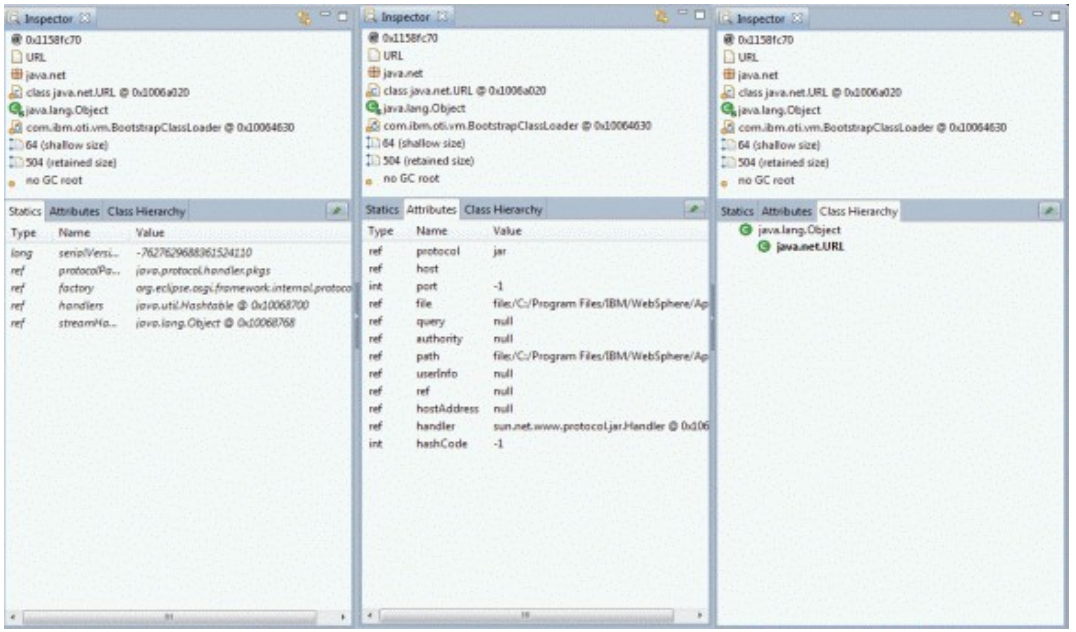
Path to GC Roots:
Provides the reference chain that prevents an object being garbage collected.



Dominator Tree grouped by Class Loader:
Lists the biggest objects using a “keep alive tree”. Grouping by Class
Loader limits the analysis to a single application in a JEE environment

Analyzing problems with Memory Analyzer

- Additional datais available in the HPROF and IBM System Dumps
  - particularly the field names and field values
- Makes it possible to diagnose a range of other problem types.
- Objects selected in Memory Analyzer have additional data show in the Inspector view
  - including the Class Hierarchy, Attributes and Statics



- This additional data is used by the IBM Extensions for Memory Analyzer

IBM Extensions for Memory Analyzer

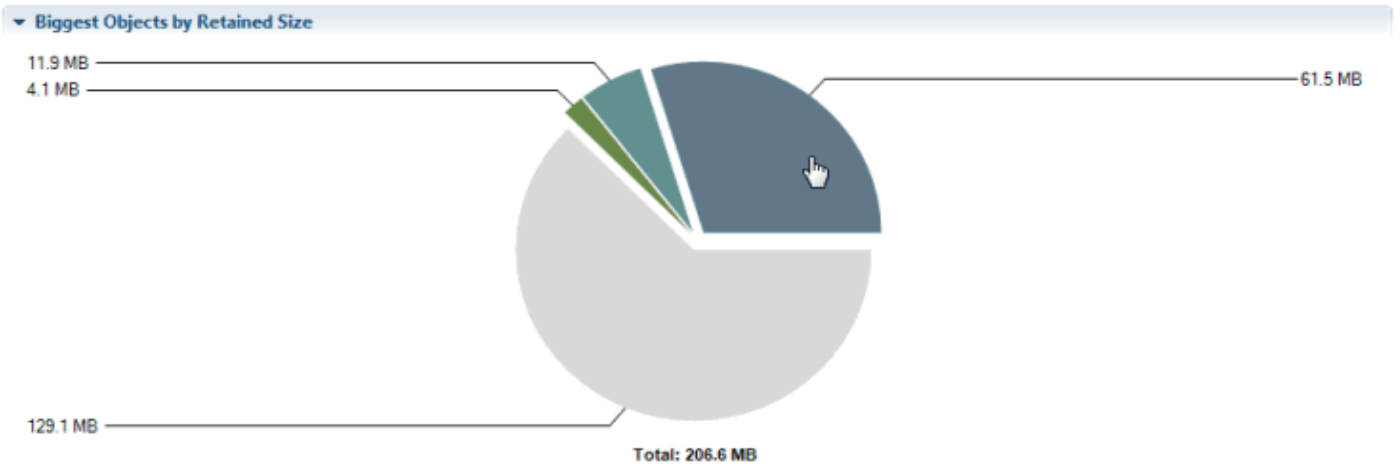
- IBM alphaWorks project to extend Memory Analyzer with IBM product knowledge
- Provides additional capabilities for debugging:
  - generic Java applications
  - WebSphere Application Server
  - WebSphere eXtreme Scale
  - CICS Transaction Gateway
  - ...and more to come
- Provides “always on” extensions, and status reports of aspects of the applications, including:
  - WebSphere Application Server Overview
  - Web Container Analysis
  - HTTP Session Analysis
  - Thread Pool Configuration and Status

Installing the IBM Extensions for Memory Analyzer

- Installed into the IBM Support Assistant (ISA) for use by Memory Analyzer
  - Can also be installed into the Eclipse Memory Analyzer Tool
- Installed into ISA using the updater
  - First needs to be configured with a new update site
    - "File → Preferences → Updater Preferences"
    - Add http://dl.alphaworks.ibm.com/ettktechnologies/updates
  - Then run "Update → Find New... → Tools Add-ons" menu option
  - Under the "JVM-based Tools" section
  - Select the plug-ins starting with: "IBM Extensions for Memory Analyzer - "
- Must install "IBM Extensions for Memory Analyzer - Utilities (required)"
  - Others are optional:
    - Java SE Runtime, WebSphere Application Server, CICS Transaction Gateway

Using the IBM extensions inside Memory Analyzer

- Always on Extensions:
  - Add information to the description of the Java objects when they are displayed in one of the Memory Analyzer views
  - Overview:



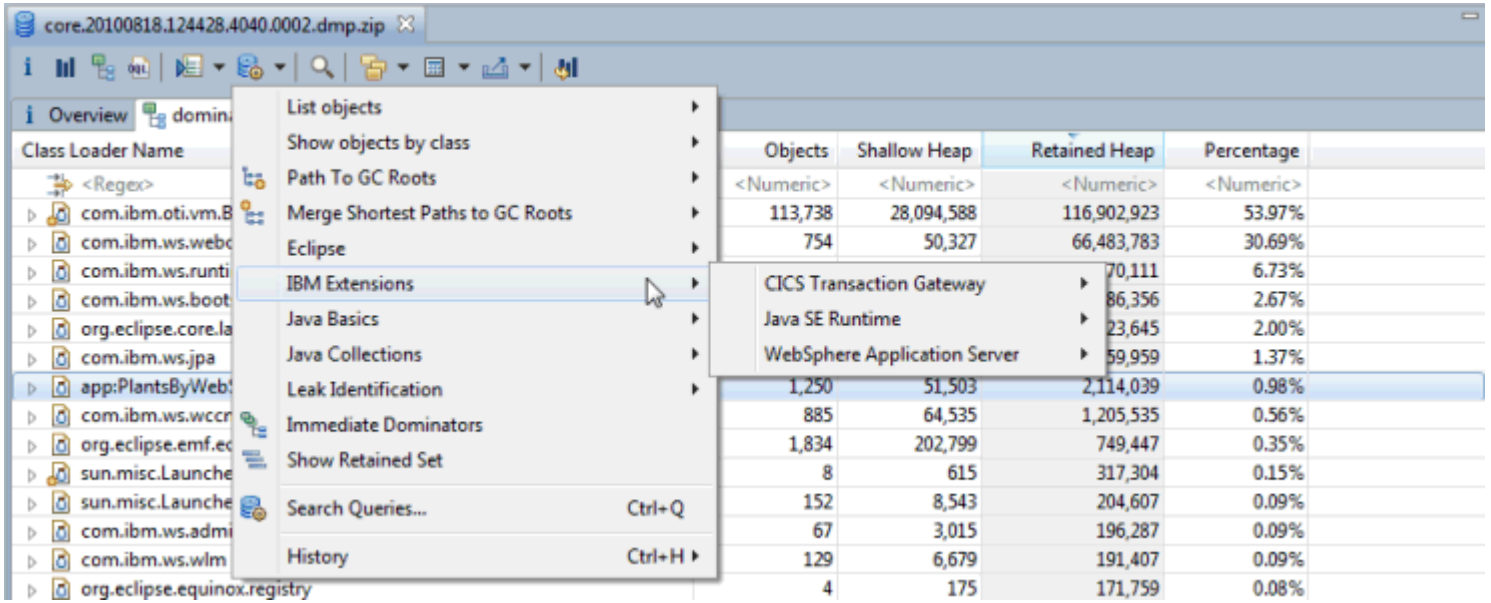
  - Dominator Tree: (by class loader)

com.ibm.ws.session.store.memory.MemoryStore @ 0x124fe928 PlantsByWebSphere at /PlantsByWebSphere currently with 1000 sessions plus 109038 overflow sessions

<Regex>	<Numeric>	<Numeric>	<Numeric>	<Numeric>
com.ibm.oti.vm.BootstrapClassLoader @ 0x10064630	113,738	28,094,588	116,902,923	53.97%
com.ibm.ws.webcontainer	754	50,327	66,483,783	30.69%
com.ibm.ws.runtime	27,149	1,330,047	14,570,111	6.73%
com.ibm.ws.bootstrap.ExtClassLoader @ 0x105354e0	1,149	67,279	5,786,356	2.67%
org.eclipse.core.launcher.Main$StartupClassLoader @ 0x1015ad00	9,068	423,455	4,323,645	2.00%
com.ibm.ws.jpa	281	16,599	2,959,959	1.37%
app:PlantsByWebSphere	1,250	51,503	2,114,039	0.98%
com.ibm.ws.wccmbase	885	64,535	1,205,535	0.56%
org.eclipse.emf.ecore	1,834	202,799	749,447	0.35%

Using the IBM extensions inside Memory Analyzer

- Query Extensions:
  - executable queries that can be found under the Query Browser pull down menu



  - Produce reports covering a number of aspects of each of the supported products

Demo

Additional Resources

- IBM Monitoring and Diagnostic Tools for Java:
  - http://www.ibm.com/developerworks/java/jdk/tools
- IBM Extensions for Memory Analyzer:
  - http://www.alphaworks.ibm.com/tech/iema
- IBM Java Runtimes and SDKs Forum:
  - http://www.ibm.com/developerworks/forums/forum.jspa?forumID=367
- IBM on Troubleshooting Java Applications Blog:
  - https://www.ibm.com/developerworks/mydeveloperworks/blogs/troubleshootingjava/?lang=en

Questions?