



# Nastel AutoPilot Heap Detective

FREE tools to prevent java memory leaks

## About Nastel AutoPilot Heap Detective

### Features and Benefits:

Feature	Benefit
Solve the leak problem in java	Quickly spot memory problems and help optimize the Java application memory footprint across multiple JVMs.
Production ready	Does not constantly run and has low overhead when used for detection.

Java provides developers with false assurance that memory leakage is something they need not be concerned with.

Quickly, spot memory problems and help optimize the Java application memory footprint across multiple JVMs

**Production ready:** does not constantly run and has low overhead when used for detection

The typical types of leads that Heap Detective can be applied to, include:

- Out of Memory
- IncreasedG Cactivity
- Heap usage
- Threads
- Handles
- JDBC statements
- ClassLoader

AutoPilot® Heap Detective™ - localhost:16400

File Edit View Window Help

Summary

Heap Usage

Class Object History

Sort Classes By: Largest Absolute Change Class Value: Total Size Apply

Snapshot Time	18 Oct 2012 10:37:21	18 Oct 2012 10:37:56	18 Oct 2012 10:38:36
Total Heap Size (bytes)	4,188,995,584	4,188,995,584 (+0)	4,188,995,584 (+0)
Free Heap Size (bytes)	3,877,710,144	3,869,449,552 (-8,260,592)	3,793,219,288 (-76,230,264)
Total Heap Objects	776,973	776,974 (+1)	789,451 (+12,477)
Total Heap References	1,022,519	1,024,022 (+1,503)	1,044,435 (+20,413)
Total Threads	82	82 (+0)	104 (+22)
Total Classes	4,972	4,972 (+0)	5,256 (+284)
Class: char[]	7,773,608	7,773,312 (-296)	8,431,456 (+658,144)
Class: byte[]	2,057,032	2,057,032 (+0)	2,231,752 (+174,720)
Class: java.lang.Object[]	6,709,216	6,709,216 (+0)	6,768,344 (+59,128)
Class: java.lang.String	2,508,600	2,508,360 (-240)	2,560,512 (+52,152)
Class: java.util.concurrent.ConcurrentHashMap\$HashEntry[]	702,160	702,160 (+0)	731,184 (+29,024)
Class: com.nastel.nfc.net.boards.FactRecord	1,036,992	1,037,184 (+192)	1,062,528 (+25,344)
Class: java.util.concurrent.ConcurrentHashMap\$HashEntry	706,208	706,208 (+0)	724,000 (+17,792)
Class: java.util.HashMap\$Entry[]	25,530,464	25,530,464 (+0)	25,544,784 (+14,320)
Class: java.lang.reflect.Method	320,160	320,160 (+0)	333,760 (+13,600)
Class: java.lang.reflect.Field	60,336	60,336 (+0)	70,560 (+10,224)
Class: java.util.concurrent.locks.ReentrantLock\$NonfairSync	367,424	367,424 (+0)	377,376 (+9,952)
Class: com.nastel.nfc.ui.admin.statustree.Sensor	200,312	200,312 (+0)	210,240 (+9,928)
Class: java.util.HashMap	160,664	160,664 (+0)	170,464 (+9,800)
Class: java.util.TreeMap\$Entry	7,920	7,920 (+0)	17,480 (+9,560)
Class: java.util.Hashtable\$Entry	535,904	535,904 (+0)	544,608 (+8,704)
Class: java.util.Hashtable\$Entry[]	217,960	217,960 (+0)	225,592 (+7,632)
Class: java.util.HashMap\$Entry	676,544	676,544 (+0)	683,008 (+6,464)

Loaded Classes: 5,256 Total Threads: 104 Heap Usage: 395,776,296/4,188,995,584 (9%) Last Snapshot: Thu 18 Oct 2012 10:38:36

This report shows a historical view of heap with the to total size of each object. The report is sorted by the largest absolute change.

**AUTOPILOT®**  
Heap Detective™

Out Of Memory,  
Increased GC Activity,  
Heap Usage,  
Threads,  
Handles,  
JDBC statements,  
ClassLoader

Heap

Heap

The screenshot shows the AutoPilot Heap Detective application interface. The main window displays a list of heap objects under the 'Object' tab. The 'Object Filter' is set to 'Top Objects By Retained Size' with a count of 10. The 'Class Filter' is set to 'com.nastel.nfc.net.boards.FactRecord'. The list of objects includes various fields and their sizes, such as 'Field: max (com.nastel.nfc.net.boards.Counter) (64 bytes)', 'Field: min (com.nastel.nfc.net.boards.Counter) (64 bytes)', 'Field: prev\_value (com.nastel.nfc.net.boards.Counter) (64 bytes)', 'Field: proxy (com.nastel.nfc.net.boards.FactProxy) (164 bytes)', 'Field: value (com.nastel.nfc.net.boards.Counter) (64 bytes)', 'Field: lastValueChange (com.nastel.nfc.net.boards.ValueChange) (122 bytes)', 'Field: history (java.util.concurrent.ArrayBlockingQueue) (24,406 bytes)', 'Field: lock (java.util.concurrent.locks.ReentrantLock) (72 bytes)', and 'Field: topic (com.nastel.nfc.net.boards.FactsBlackBoard) (2,299,908 bytes)'. The 'Field: history' object is highlighted in blue. To the right of the list, a table shows the properties of the selected object:

Item	Value
Type	Object Instance
Size	64 bytes
Retained Size	24,406 bytes
References To	1
Objects Referred To	4
Field Name	history
Defined in	com.nastel.nfc.net.boards.FactRecord

Below the table, a 'References To' section shows a tree view of references to the selected object, including a 'FIELD' reference to 'Field: history'. The bottom status bar shows 'Loaded Classes: 5,256', 'Total Threads: 104', 'Heap Usage: 395,776,296/4,188,995,584 (9%)', and 'Last Snapshot: Thu 18 Oct 2012 10:38:36'.

This shows the "top objects by retained size" within heap. Users can interact with this view and highlight a heap object, see its properties such as retained size, the fields it is comprised of and all the references it has to objects within the JVM

**Disclaimer:**

Information contained in this data sheet is up-to-date and correct as at the date of issue. All product, product specifications and data are subject to change without notice to improve reliability and function of our software. To the maximum extent permitted by law, Nastel Technologies, Inc. will not be responsible for damages of any nature resulting from the use or reliance upon the information contained in this data sheet. No express or implied warranties are given other than those implies mandatory by law.