Do you know who did it?
Troubleshooting issues in
WebSphere Application Server

Part-4: Garbage, Memory leak and Out of memory

Joseph Amrith Raj

http://facebook.com/MiddlewareLibrary

http://gplus.to/WebSphereLibrary

http://about.me/WebSphereLibrary

Table of Contents

Fourth part of this series [do you know who did it? Troubleshooting issues in websphere application server], we will be discussing how to obtain and diagnose Garbage collection and Memory related issues in WebSphere application server using Garbage collection and Memory Visualizer.

About GC and Memory Visualizer

Garbage Collection and Memory Visualizer is part of the IBM support assistant workbench. This tool can visualize the garbage collection data from IBM JRE 1.4.2 or higher. This tools helps you get insights into garbage collection using graphs with data like Heap size, used heap size, pause times,

Using GC and Memory Visualizer

The GC and Memory Visualizer allow you to visualize your garbage collection data, as raw datasets, line plots, reports, and images. The "Line plot" portion of the tool displays graphs of over forty different garbage collection data characteristics – including used heap, pause times, and the reason for garbage collection being triggered. You can read in multiple sets of garbage collection logs and display them together on a single set of axes, which allows you to easily compare garbage collection behavior across multiple test runs of an application. The "Report" section of the tool contains a summary of the line plot data, along with information on general garbage collection behavior and heap sizing recommendations based on overall heap occupancy.

The tool can provide the following:

- a graphical display of a wide range of verbose GC data values
- tuning recommendations and detection of problems such as memory leaks
- report, raw log, tabulated data and graph views
- saving of data to HTML reports, jpeg images or .csv files (for export to spreadsheets)
- viewing and comparing multiple logs
- analysis of optthruput, optavgpause, and gencon GC modes

Obtaining and installing the tool

- Start the IBM support assistant tool
- Go to update → find new → tools add-ons
- Expand he JMV based tools from the results and select "IBM Monitoring and Diagnostic Tools for Java – Garbage collection and Memory Visualizer" and proceed to install.
- After installation is complete, restart ISA.

To use this tool, you need to have the GC logging information, which can be collected by enabling verbose GC.

Enabling Verbose GC

In the WebSphere Application Server administration console:

- Servers → server_name → Server Infrastructure → Java Process Management →process definition → Additional properties →Java Virtual Machine and enable "Verbose Garbage Collection" check box

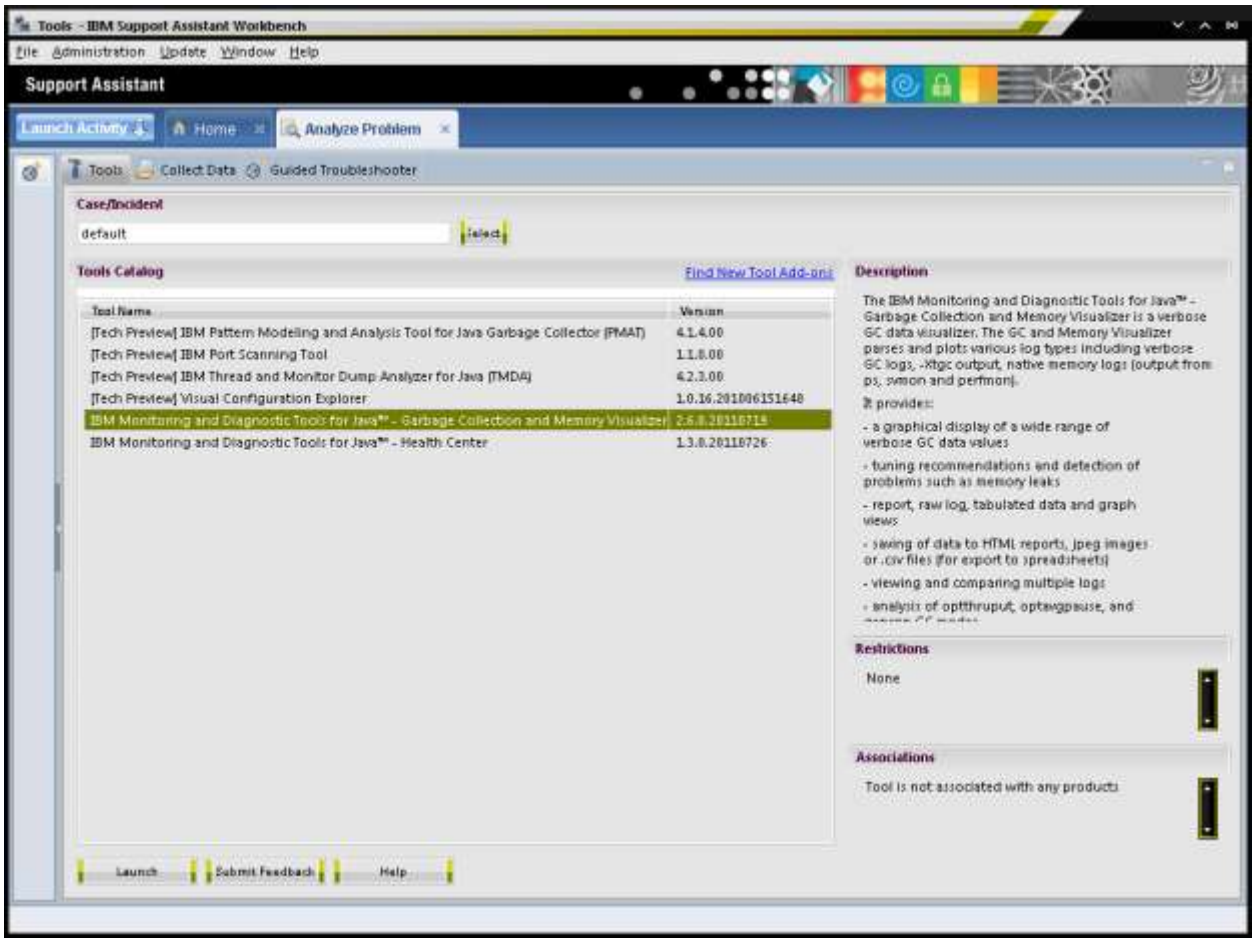
- Restart the server on which verbose GC is enabled.

Output will be written to the native_stderr. Log

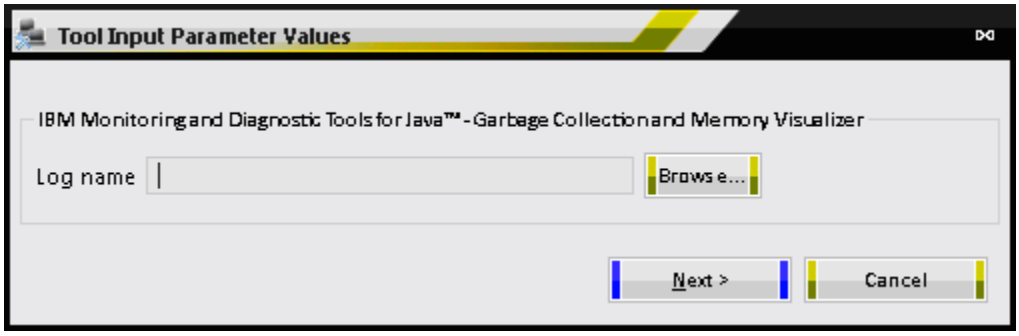
Analyze Verbose GC log

Start the ISA and go to Analyze problems

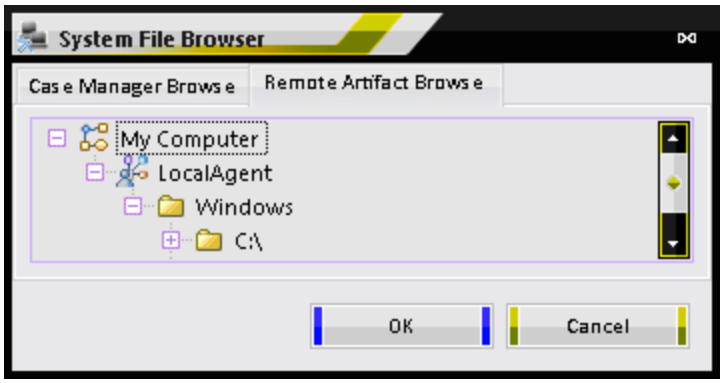
Select the tool in the list and click launch



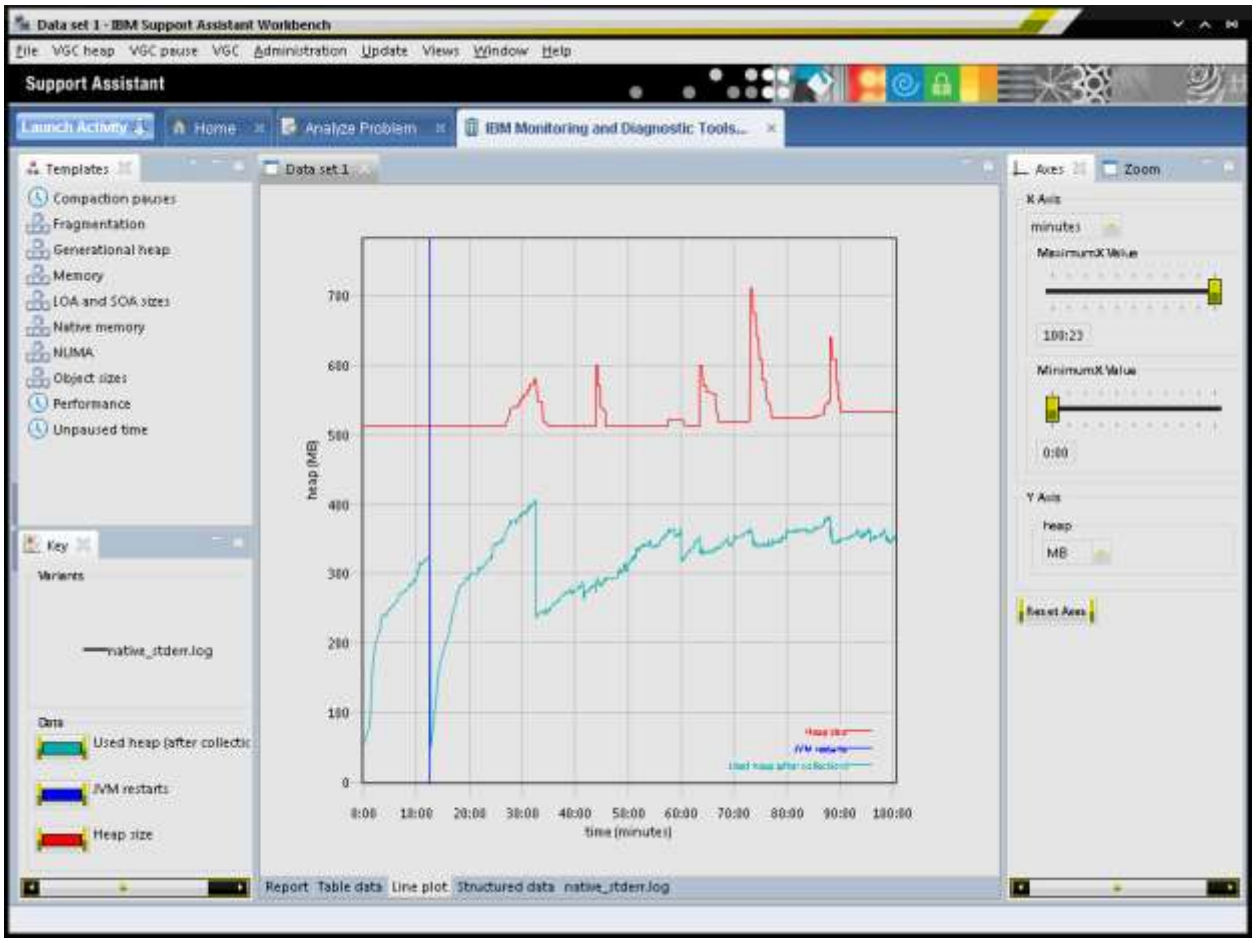
A pop-up window will ask you to provide the Log file which has the Garbage collection data.



Click on Browse → Remote Artifact Browse and select the log file



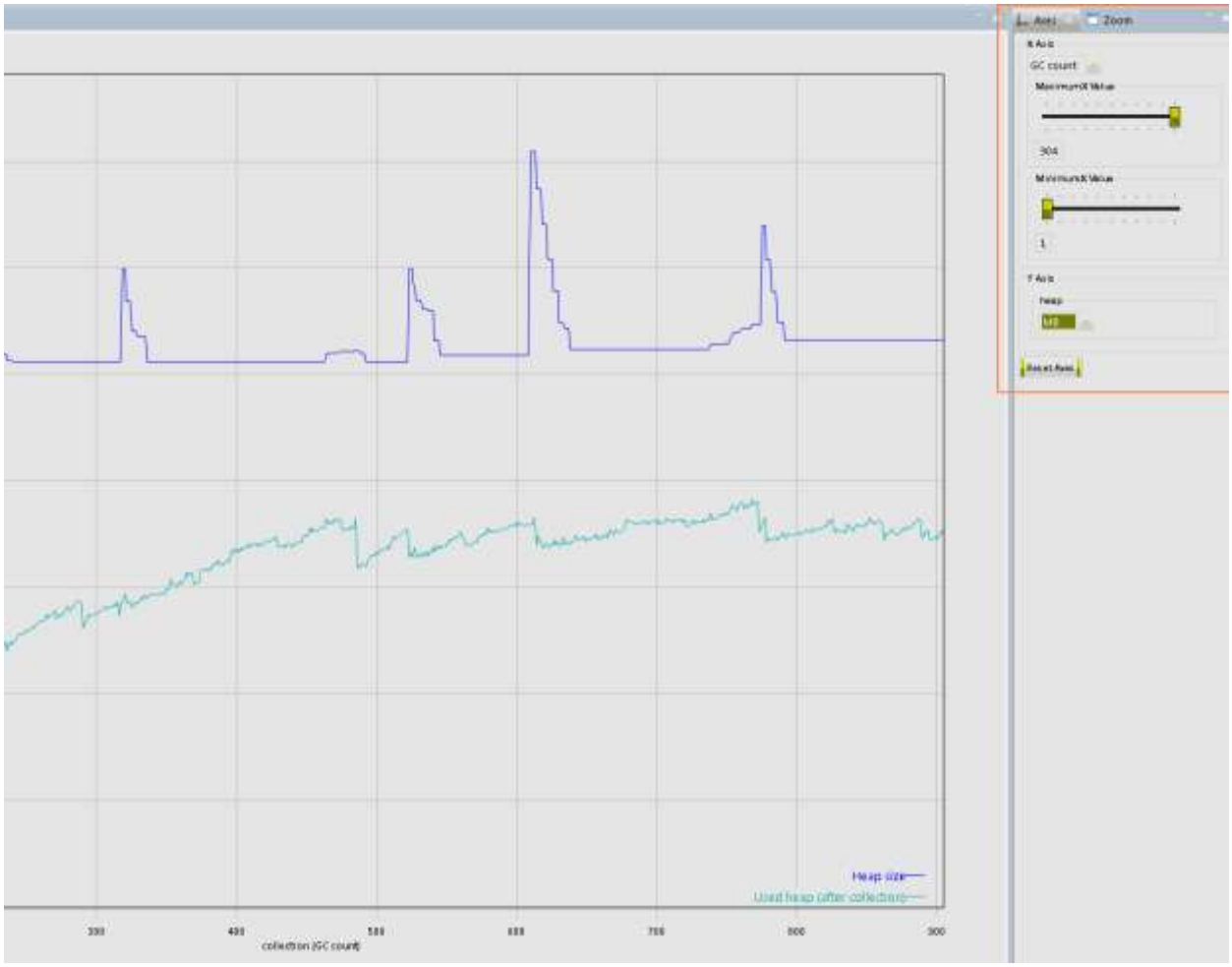
Click next and the tool will automatically analyze the log file and shows you data set from the input log file. The default setting is to show the line plot.



Customizations and Navigation

You can select the X and Y axis parameters for the line plots (graphs). Check on the right side panel of the line plot. You will find Axes tab and Zoom tab.

Axes Tab



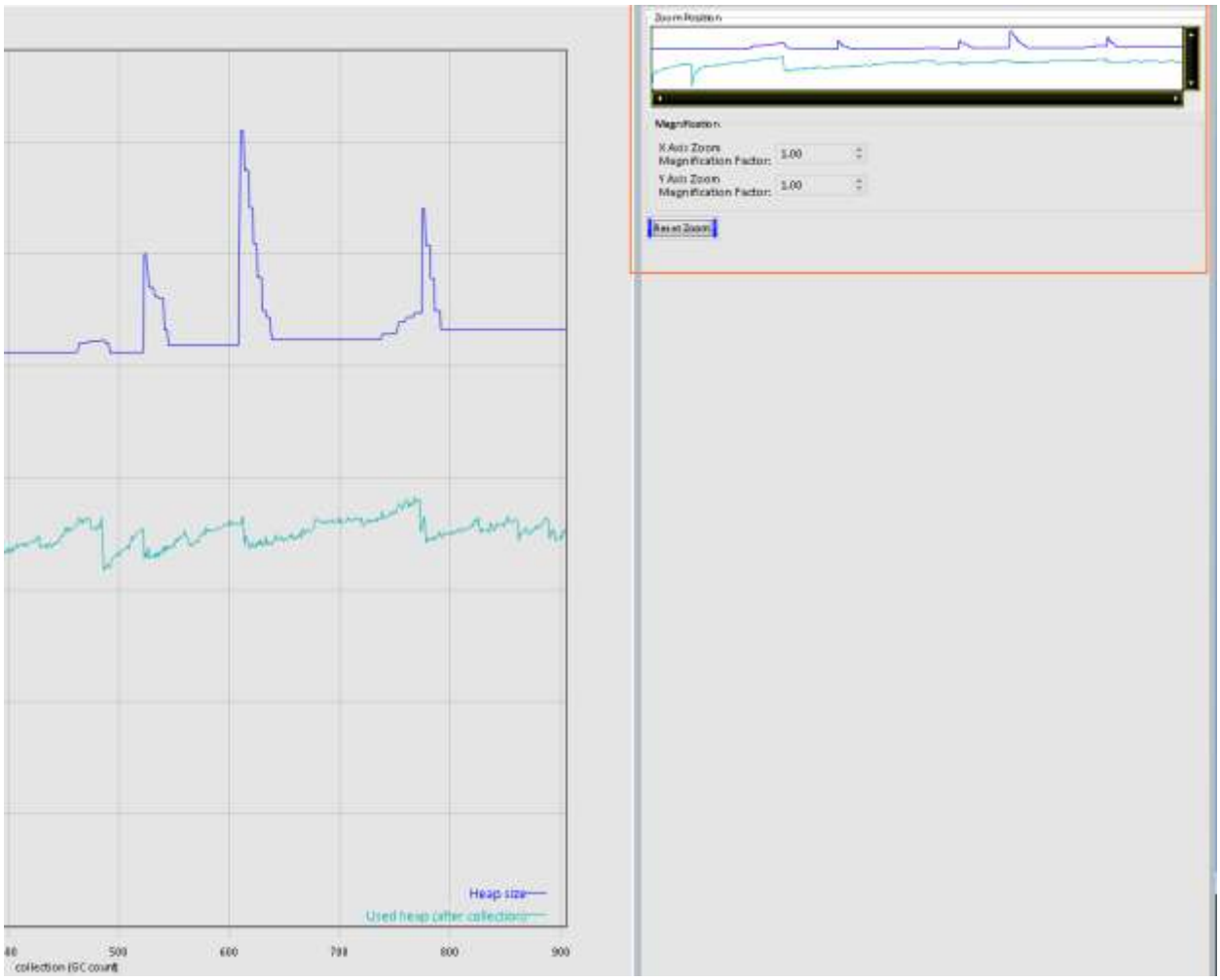
Use this tab to customize the X and Y axes parameters.

X Axis is generally time which can be customized to msec, seconds, minutes, hours, days. And X axis can be set to GC count as well.

Y axis is generally Size, which can be customized to bytes, KB, MB, GB and percentage.

Zoom tab

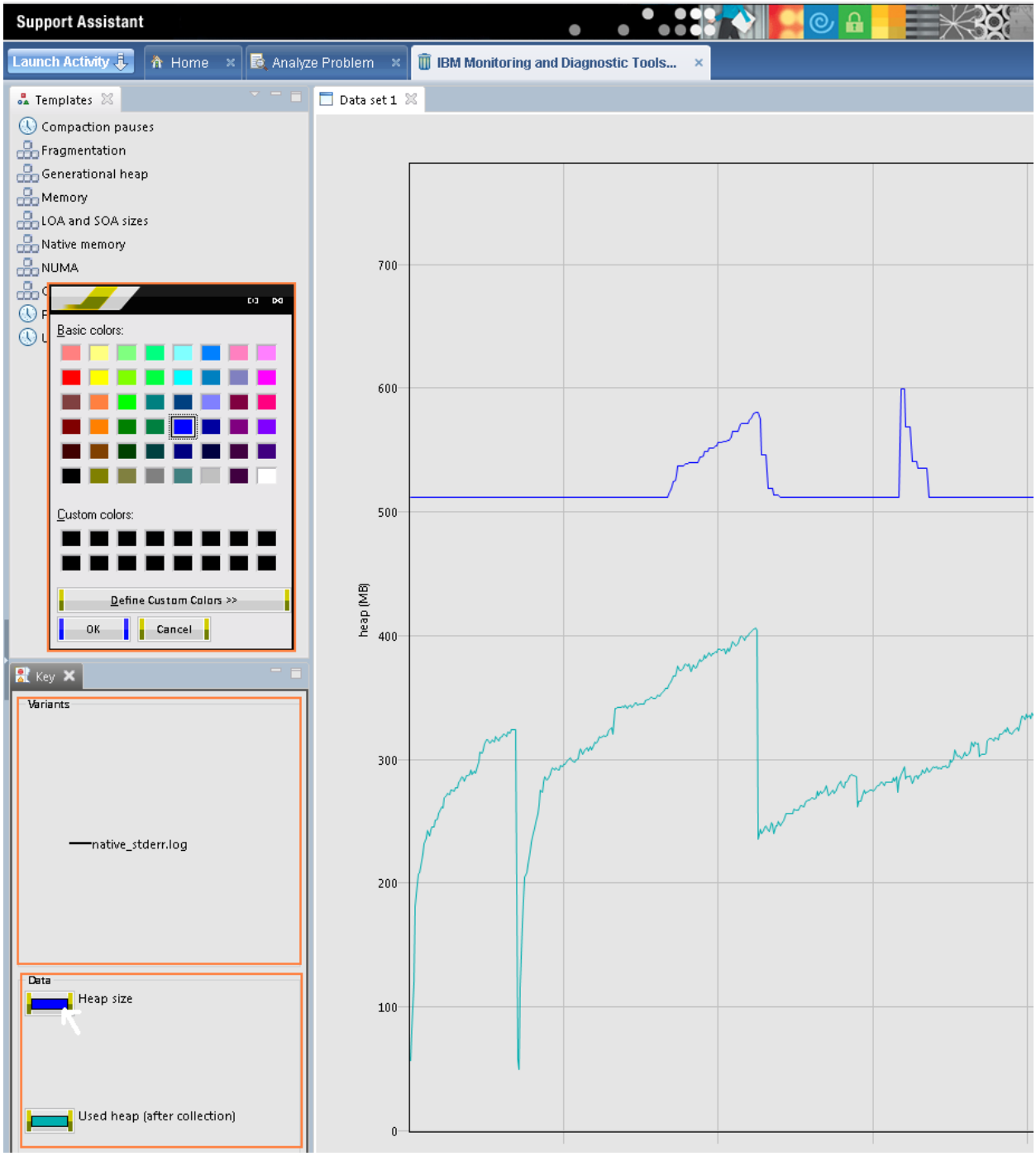
Use this to customize the zooming on either x-axis or y-axis or both.



The other way to zoom is to select area of interest on the graph, by holding the mouse left-click.

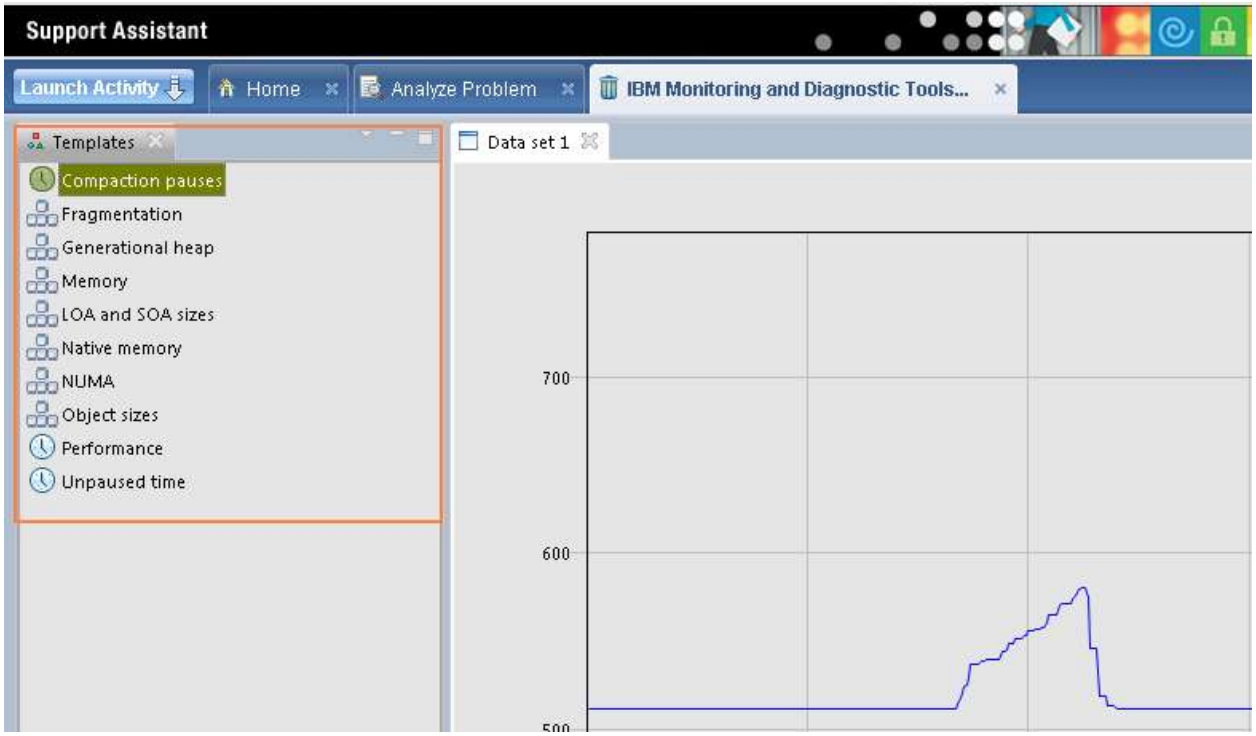
Keys panel

The keys panel shows what variants and data are being displayed in the graph. You can customize the colors of the data on the graph. Click on the data being shown the data panel to change color of that data on the graph.



Templates

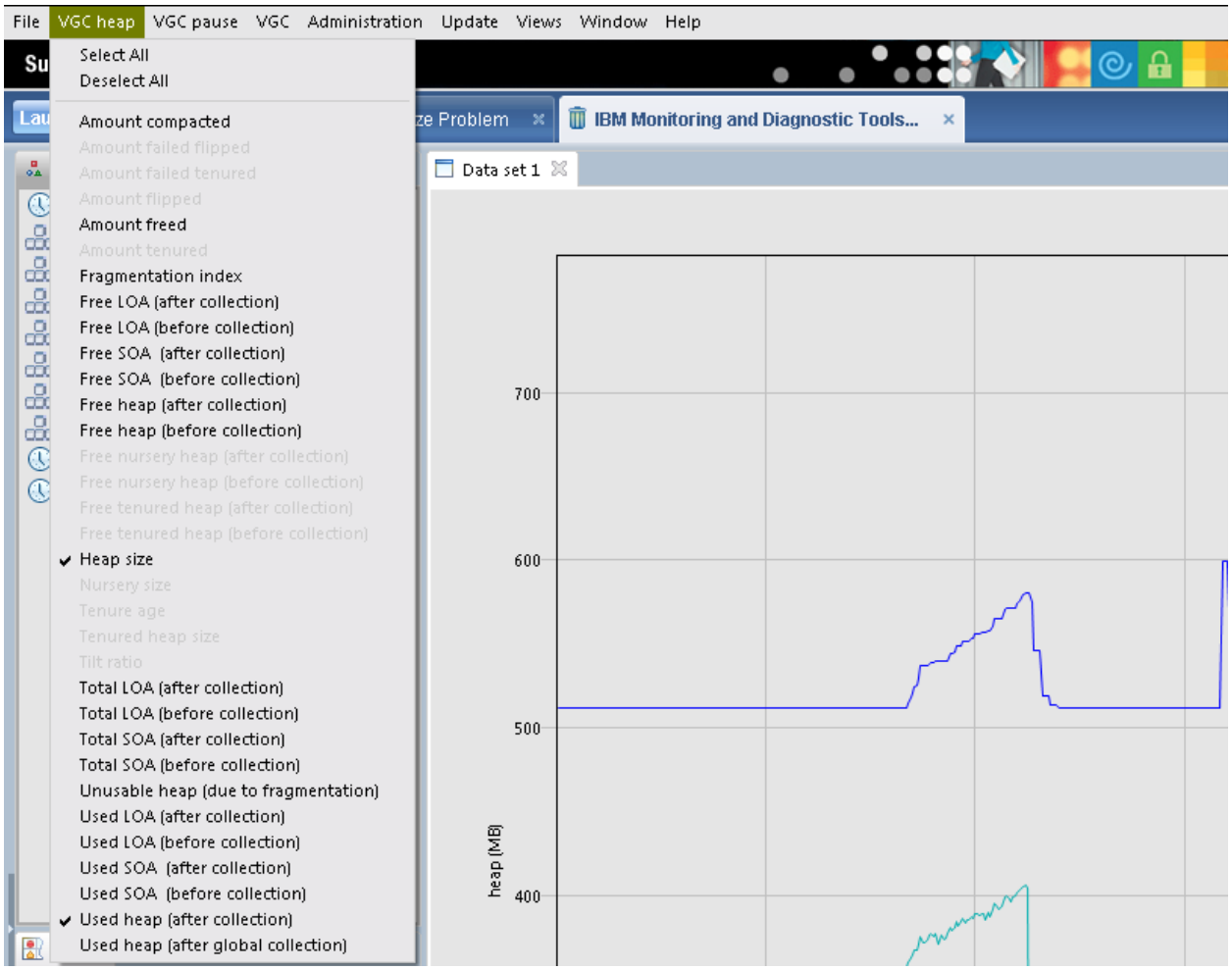
This panel lists the templates available to plot the graphs. Selecting [double-click]a template from the list shows the relevant graph.



Menu and Customizations

Different options are available on the menu to plot the customized graphs

Clicking in VGC heap/ VGC pause will display the available data options


File
VGC heap
VGC pause
VGC
Administration
Update
Views
Window
Help
Select All
Deselect All
Amount compacted
Amount failed flipped
Amount failed tenured
Amount flipped
Amount freed
Amount tenured
Fragmentation index
Free LOA (after collection)
Free LOA (before collection)
Free SOA (after collection)
Free SOA (before collection)
Free heap (after collection)
Free heap (before collection)
Free nursery heap (after collection)
Free nursery heap (before collection)
Free tenured heap (after collection)
Free tenured heap (before collection)
Heap size
Nursery size
Tenure age
Tenured heap size
Tilt ratio
Total LOA (after collection)
Total LOA (before collection)
Total SOA (after collection)
Total SOA (before collection)
Unusable heap (due to fragmentation)
Used LOA (after collection)
Used LOA (before collection)
Used SOA (after collection)
Used SOA (before collection)
Used heap (after collection)
Used heap (after global collection)
IBM Monitoring and Diagnostic Tools...
Data set 1
700
600
500
400
heap (MB)

Analyze and interpret the data

The following options can be used to understand the data from the analyzed VGC log.

Report | Table data | Line plot | Structured data | native_stderr.log

Report: By default report shows tuning recommendations, summary heap size and used heap size graphs. You can customize the report using the VGC menu item. The other options in the VGC menu for report are GC reason, Classes loaded, JVM restarts etc...

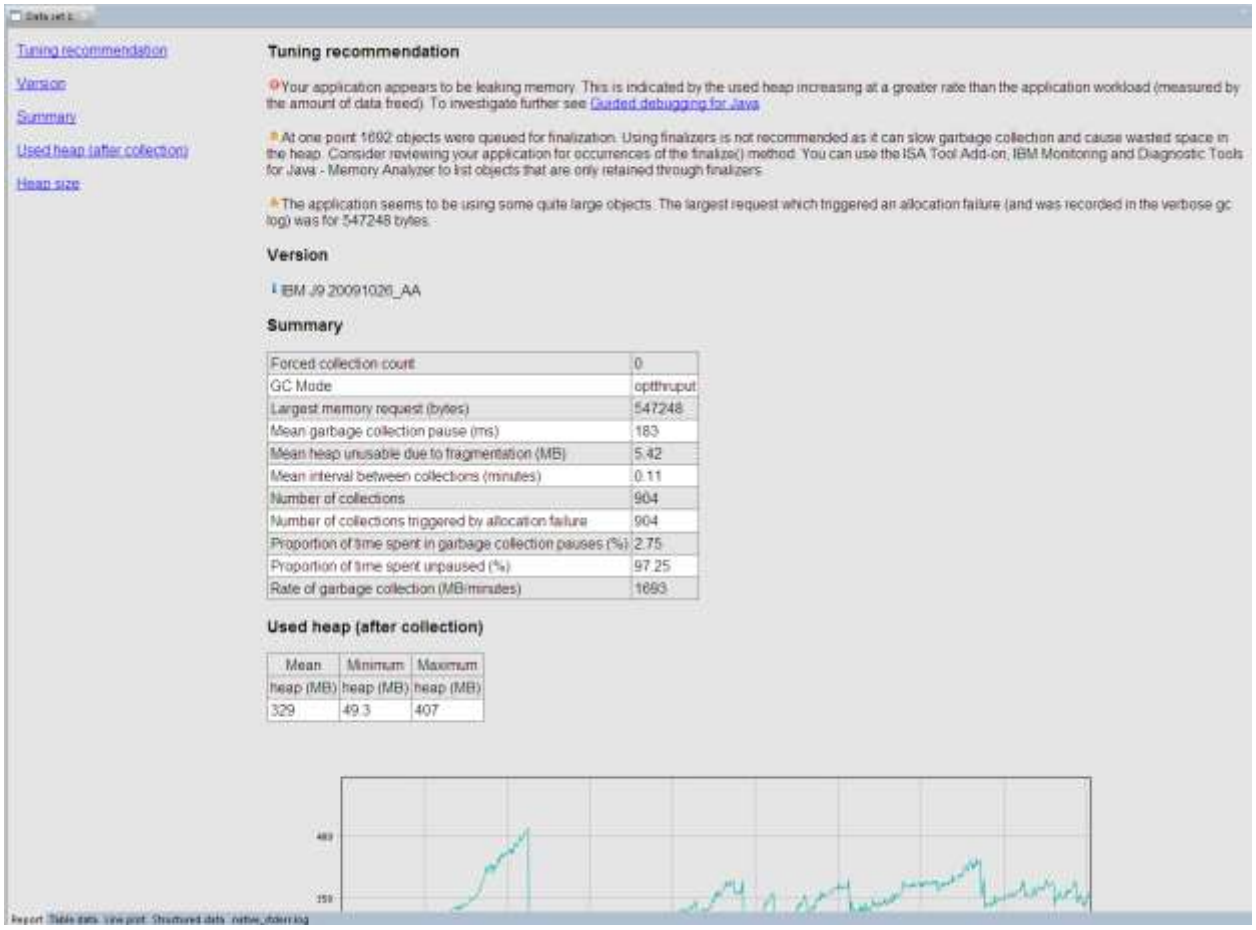
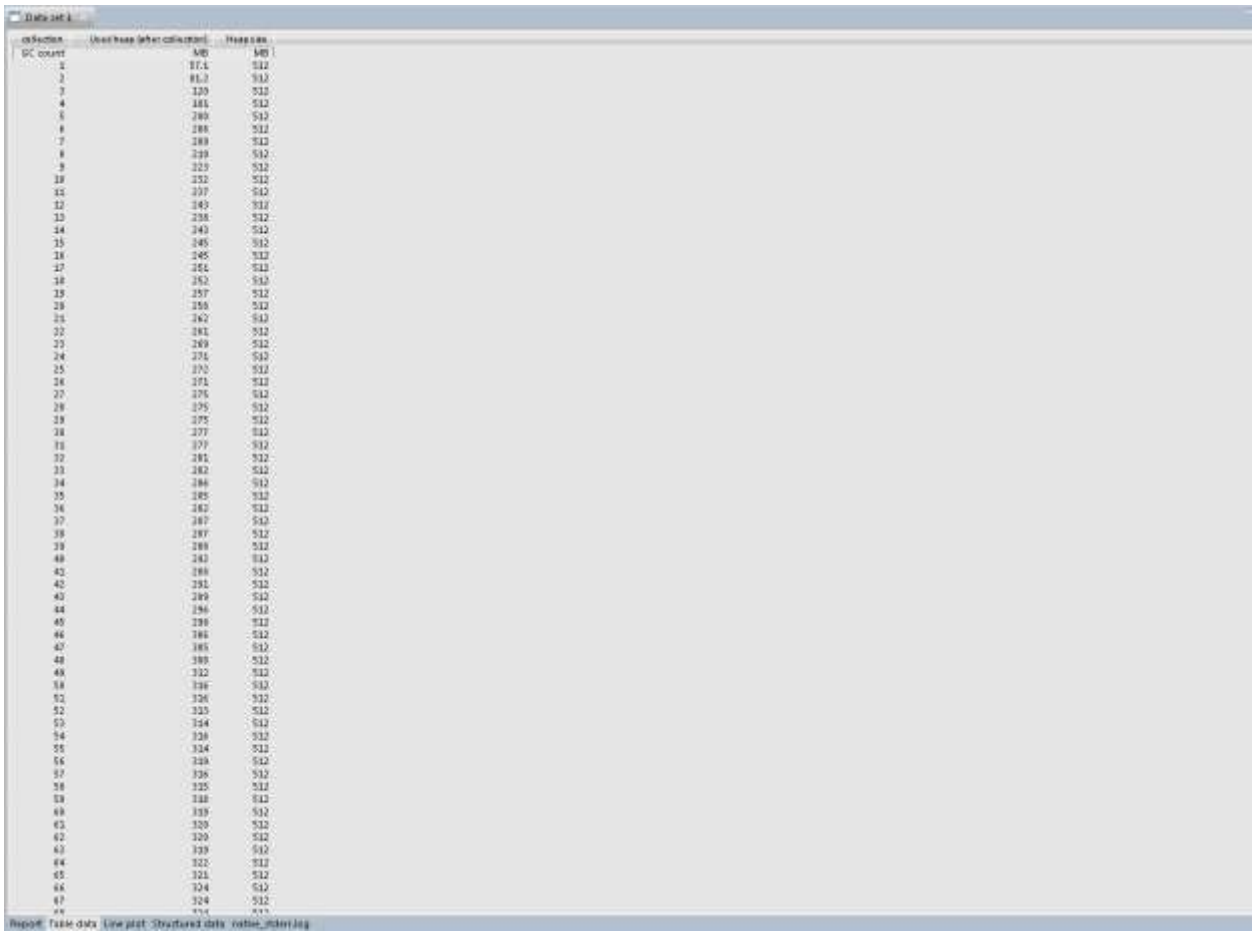
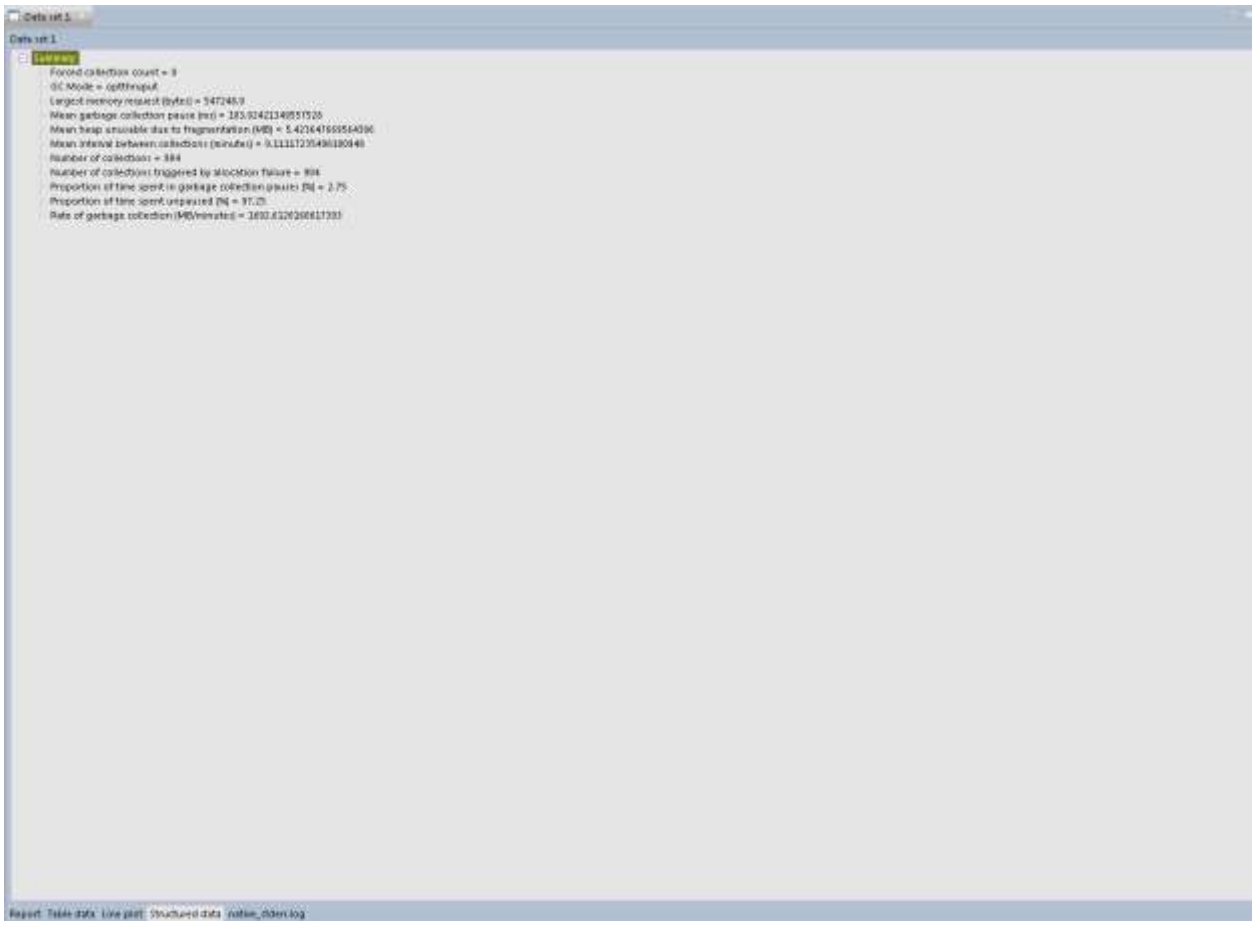


Table data: Shows data from the GC log in table format. Defaults are GC count, used heap and Heap size. You can customize this tabular representation of the data by selecting other data parameters from either VGC pause or VGC heap menu items.



Structured Data: shows the summary from the analyzed GC log.



Line plot: shows the graphical representation of the data using the selected template or customized options.

Cool … any real-time usage or examples?

- You can use this tool to diagnose Memory Leaks. If the used heap line is creeping up when there's no obvious reason for the memory requirements of the application to be increasing, there may be a leak. The GC and Memory Visualizer looks for this pattern and adds a comment to the tuning recommendation if it detects something that is likely to be a leak. However the tool will not show you which object are causing the leak.
- You can use this tool to determine what will be optimized Heap size for your applications. If the heap is so small that the data required by the application will not fit into it, then the application will run out of memory and terminate with an OutOfMemoryError. If the heap has room for the application data but not much room to spare, the garbage collector will have to spend a lot of time ensuring that there is room in the heap for new allocations, and this will hurt application performance. A heap that is too big usually won't have a negative effect on application performance, but it is wasteful, and GC pauses may be long.
- You can also use this tool to estimate the application throughput and response times. You can change the GC policy to different available options and analyze which policy is giving best response times. But it always recommended collecting the GC data when the application is under good load.

References

1. Garbage Collection policies: part-1 and Part-2
2. Memory leak detection and analysis : part-1 and part-2
3. Do you know who did it? Troubleshooting issues in websphere application server Series