

RJam Analyzer

Table of Contents

Table of Contents	1
Introduction.....	2
1. Quick Start (Run your first report).....	2
2. Select the reports.....	3
3. Report Anatomy.....	4
Graph View.....	5
Comments	5
Data View	7
4. Defining Selection Criteria	8
Adding Detail to a report query	9
5. Navigation Tree	10
6. Export Data.....	11
7. Favorites.....	13
8. Thread Analysis	14
9. Statistics	15
10. Available Reports.....	16
Http Activity	16
Jdbc Activity.....	16
Gc Analysis.....	17
Gc Regression	18
JVM CPU.....	19
Errors.....	20
Long Running Transactions.....	21
Method Hangs.....	22
11. Auto Refresh	23
12. Data Cache (help menu).....	24

Introduction

The RJam field agent resides in the application runtime environment and collects performances statistics as the application runs and stores the data in a centralized database. Once the data is captured, we can use the RJam Analyzer to view the data and gain some insight into the internal health of the application.

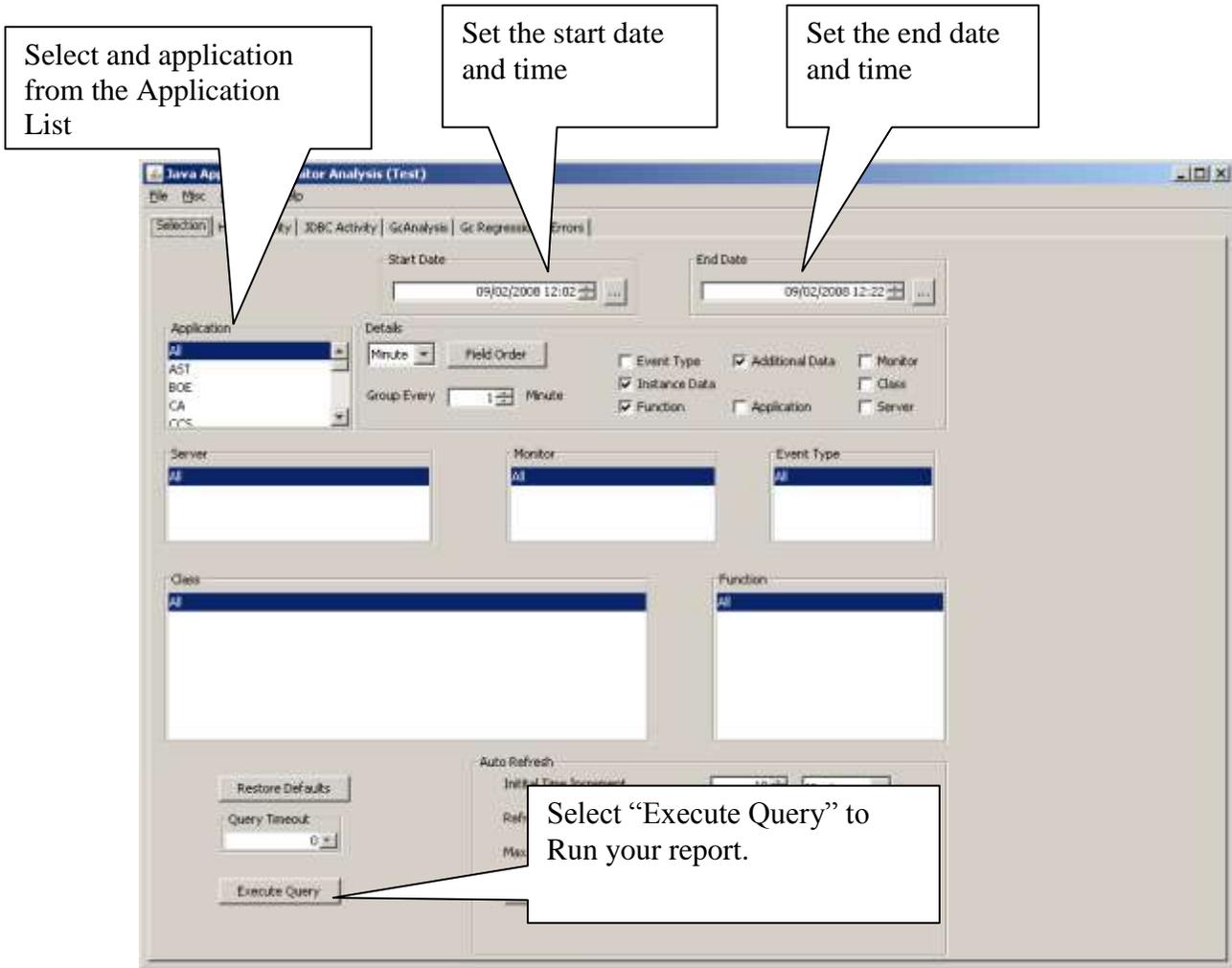
There is no direct link between this tool and the RJam collection agent installed in the application.

1. Quick Start (Run your first report)

To run your first report, select the application that you would like to work with, set the start and end times and press the “Execute Query” button. This will populate the default reports with the last 20 minutes of data for that application.

When the analysis tool starts up, the start and end dates are set to the last 20 minutes and the “Application List” is populated with all of the available applications.

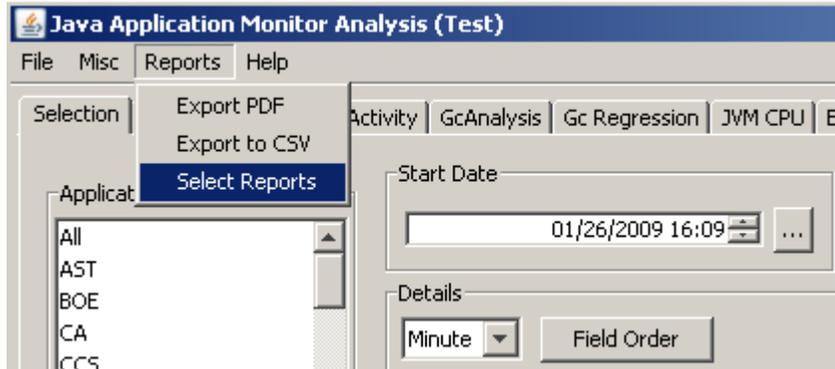
To populate any of the other lists (Server, Monitor, Event Type, Class and Function), click on the list with your mouse and the analysis tool will populate them with any data that is relevant to the selected application.



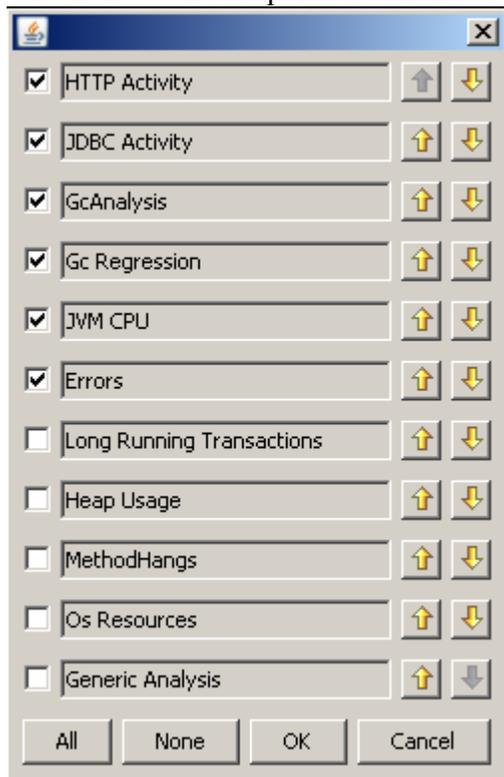
2. Select the reports

The Monitor Analysis tool allows you to generate a number of reports to view the data. You can select the Reports that you are interested in from the 'Reports' menu.

You can run as many reports at a time as you like. If you are reporting on a large amount of data, you may need to reduce the number of reports to conserve memory. This might be the case if you are reporting on a very active application or if you are reporting on a long time frame.



Check the desired reports the click 'OK'



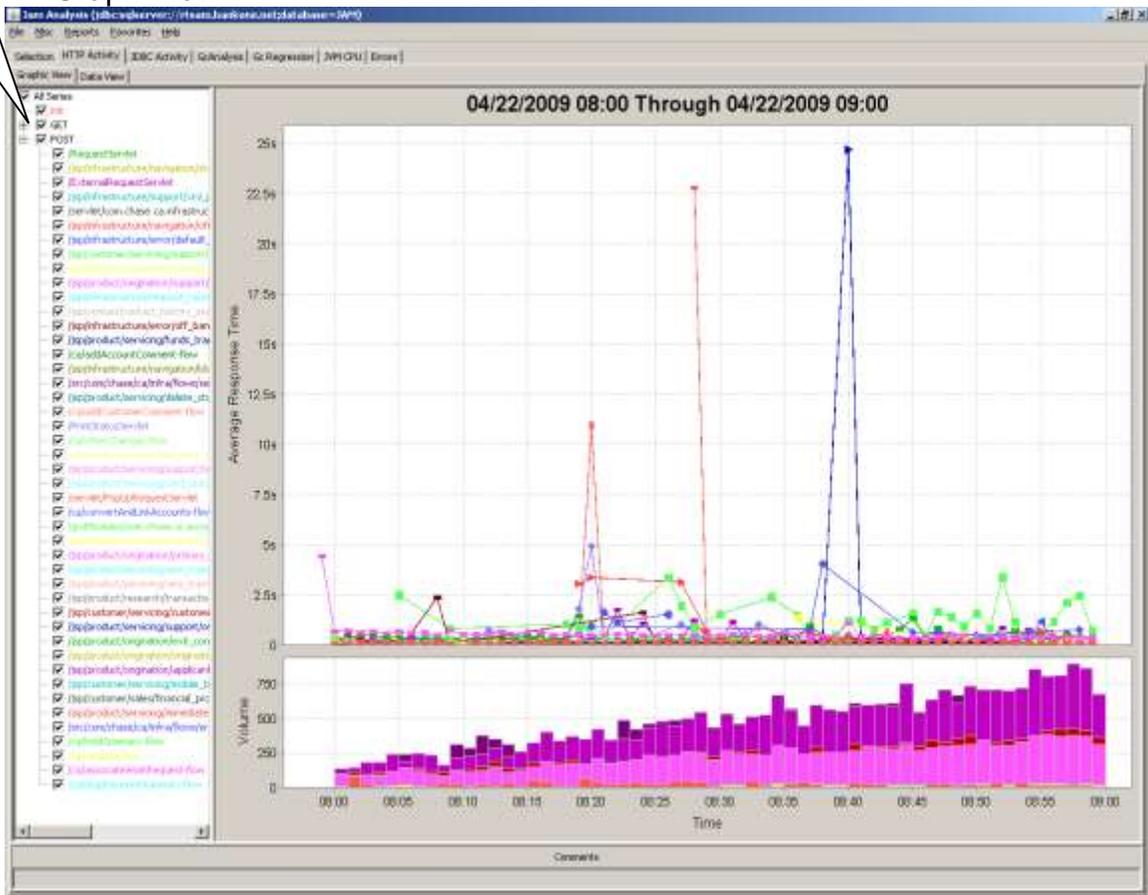
3. Report Anatomy

The reports are intended to help understand the data by segregating data into logical groups and attempting to highlight common problem areas. While the data in each report is different, they share a common format.

Each report has two views, the Graph View and the Data View.

Navigation Tree
(Manage Visibility)

Graph View



The Graph view has three major sections;

- **Navigation Tree (Legend):** In addition to listing the data series on the report, the legend helps you overcome the “Information Overload” created by the sheer amount of data. Use the Legend to show or hide any of the data series on the report.
- **Average Response Time:** This is a line chart showing the average response time for each data series. The response time is an average of all data points for that series based on the grouping specified in the selection criteria (Second, Minute, ...).
- **Volume:** This is a bar chart showing the number of transaction (data points) for each series.

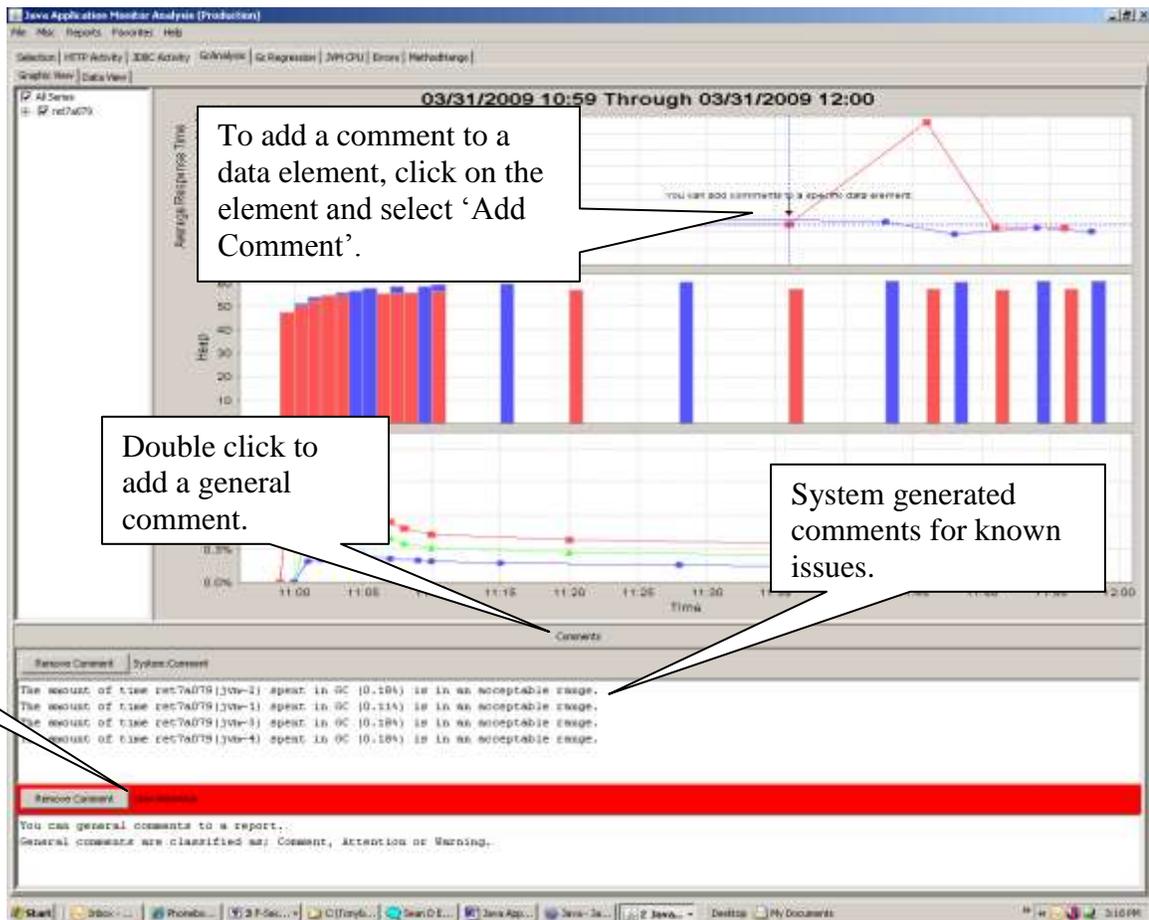
Comments

Comments are used to point out or bring attention to specific data points (mouse click on the data point and select ‘Add Comment’).



System generated comments are designed to point out areas of known problems (such as garbage collection performance problems).

User comments are comments that you add to your own reports. This can be very helpful to draw attention to specific data when exporting to a PDF file.



Data View

The data view shows the raw data used to generate the report. The data will vary by report, but here are the common columns.

- TxCount: Represents the “Transaction Count” and is used to populate the Volume chart in the Graph view.
- Min, Max and Ave: Represent the min max and average response time in milliseconds (1000 = one seconds).
- Method: This is the name of the method or the logical function.
- Start Data: The start data is specific the function and is defined by the “Collector” used to collect the data.
- End Data: The end data is also specific to the “Collector” used to collect data. It is normally empty except for error when is will contain a description of the Exception.
- Date: This is the date and time of the data point.

TxCount	Min	Max	Ave	Method	Start Data	End Data	Date
28	0.0	0.0	0.0	executeQuery			2011/08/30 16:56
10	0.0	0.0	0.0	executeQuery			2011/08/30 16:57
26	0.0	0.0	0.0	executeQuery			2011/08/30 16:58
9	0.0	0.0	0.0	executeQuery			2011/08/30 16:59
36	0.0	0.0	0.0	executeQuery			2011/08/30 17:00
46	0.0	0.0	0.0	executeQuery			2011/08/30 17:01
6	0.0	0.0	0.0	executeQuery			2011/08/30 17:02
8	0.0	0.0	0.0	executeQuery			2011/08/30 17:03
16	0.0	0.0	0.0	executeQuery			2011/08/30 17:04
18	0.0	0.0	0.0	executeQuery			2011/08/30 17:05
12	0.0	0.0	0.0	executeQuery			2011/08/30 17:06
339	0.0	76.0	19.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:46
363	0.0	63.0	23.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:46
362	0.0	63.0	20.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:46
394	0.0	63.0	17.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:50
374	0.0	65.0	19.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:51
322	0.0	78.0	20.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:52
750	0.0	63.0	18.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:52
332	0.0	63.0	11.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:52
375	0.0	63.0	13.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:56
366	0.0	63.0	21.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:56
654	0.0	63.0	18.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:57
396	0.0	63.0	17.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:58
491	0.0	63.0	11.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 16:59
386	0.0	63.0	17.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 17:00
732	0.0	63.0	17.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 17:01
369	0.0	78.0	18.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 17:02
390	0.0	63.0	17.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 17:02
369	0.0	63.0	17.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 17:04
778	0.0	63.0	18.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 17:05
368	0.0	63.0	20.0	executeQuery	SELECT * FROM CONFIGURATION WHERE ID = ?		2011/08/30 17:06
2	0.0	0.0	0.0	executeQuery	select * from TRANSACTION where status ...		2011/08/30 16:46
2	0.0	16.0	8.0	executeQuery	select * from TRANSACTION where status ...		2011/08/30 16:50
2	0.0	0.0	0.0	executeQuery	select * from TRANSACTION where status ...		2011/08/30 16:51
2	0.0	0.0	0.0	executeQuery	select * from TRANSACTION where status ...		2011/08/30 16:52
2	0.0	0.0	0.0	executeQuery	select * from TRANSACTION where status ...		2011/08/30 16:52
2	0.0	0.0	0.0	executeQuery	select * from TRANSACTION where status ...		2011/08/30 16:52
2	0.0	16.0	8.0	executeQuery	select * from TRANSACTION where status ...		2011/08/30 17:00
2	0.0	0.0	0.0	executeQuery	select * from TRANSACTION where status ...		2011/08/30 17:01
2	0.0	0.0	0.0	executeQuery	select * from TRANSACTION where status ...		2011/08/30 17:04

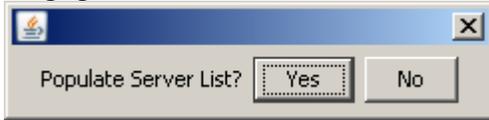
4. Defining Selection Criteria

Using the “Selection” page you can define the criteria you want to include in you reports. In most cases, you will only need to specify the time frame and the application. However, you can also filter the report using the Server, Class, Function (or Method), Monitor and Event Type.

- **Start Date / End Date:** The start and end date fields specify the date and time of the reports.
- **Application:** Select the application(s) to include in these reports. Application names are defined during the installation process. While you can select multiple applications, as a general rule, you’ll want to restrict your reports to a single application.
- **Server:** Select specific server names to restrict the reports to data collected from those servers (only the short DNS names).
- **Class:** Select specific class names to restrict reports to only those classes. The fully qualified name is available.
- **Function (Method):** Select specific functions name to restrict the reports to only those functions.

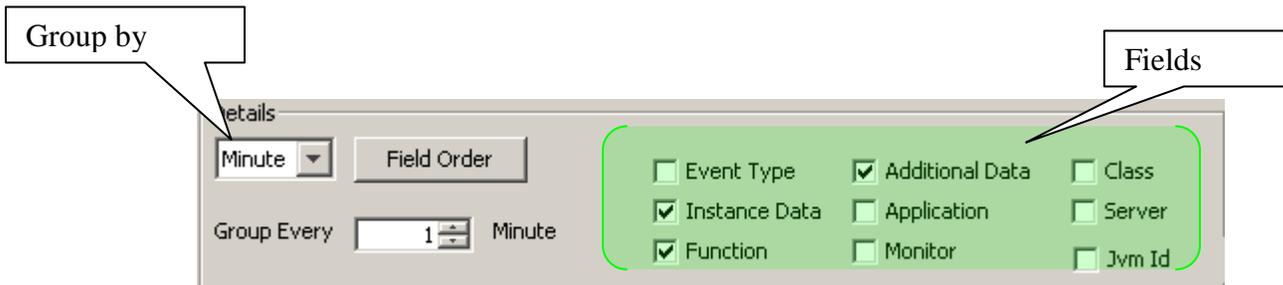
The screenshot shows the 'Application Monitor Analysis (Test)' window. It features several filter sections: 'Application' (with a dropdown menu), 'Server' (with a dropdown menu), 'Class' (with a dropdown menu), 'Function' (with a dropdown menu), 'Monitor' (with a dropdown menu), and 'Event Type' (with a dropdown menu). There are also checkboxes for 'Event Type', 'Instance Data', 'Function', 'Additional Data', 'Application', 'Monitor', 'Class', and 'Server'. A 'Details' section includes a 'Minute' dropdown, a 'Field Order' button, and a 'Group Every' field. At the bottom, there are 'Auto Refresh' settings for 'Initial Time Increment', 'Refresh Time Increment', and 'Max Age', along with 'Restore Defaults', 'Query Timeout', 'Execute Query', and 'Begin Auto Refresh' buttons. Callouts point to various elements: 'Filter by application name.' points to the Application dropdown; 'Filter by server name.' points to the Server dropdown; 'Filter by Java class name' points to the Class dropdown; 'Filter by Monitor' points to the Monitor dropdown; 'Filter by Function or Method name.' points to the Function dropdown; 'Filter by Event Type' points to the Event Type dropdown; and 'Add detail to the reports.' points to the checkboxes in the Details section.

To populate the selection fields with data, select the desired list and then select “yes”.

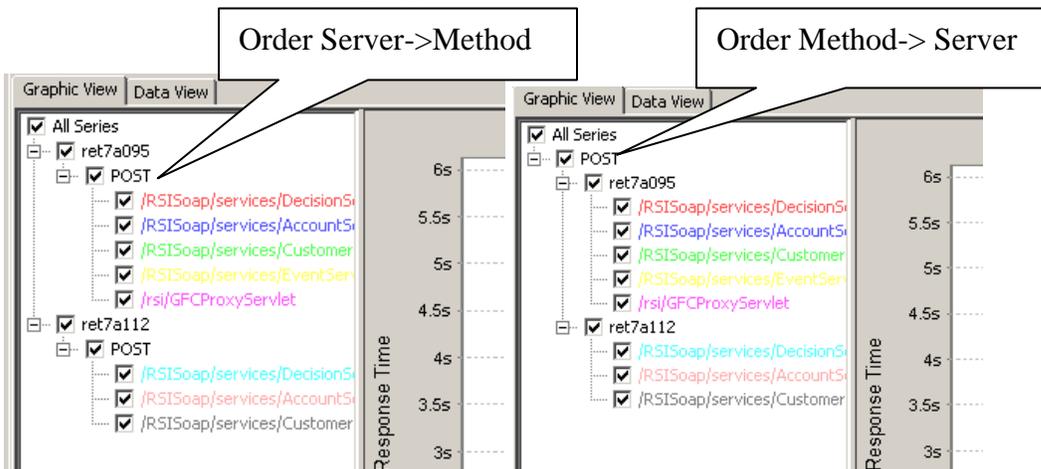


Adding Detail to a report query

The detail section of the Selection page let’s you control how much (or how little) detail is included in your reports.



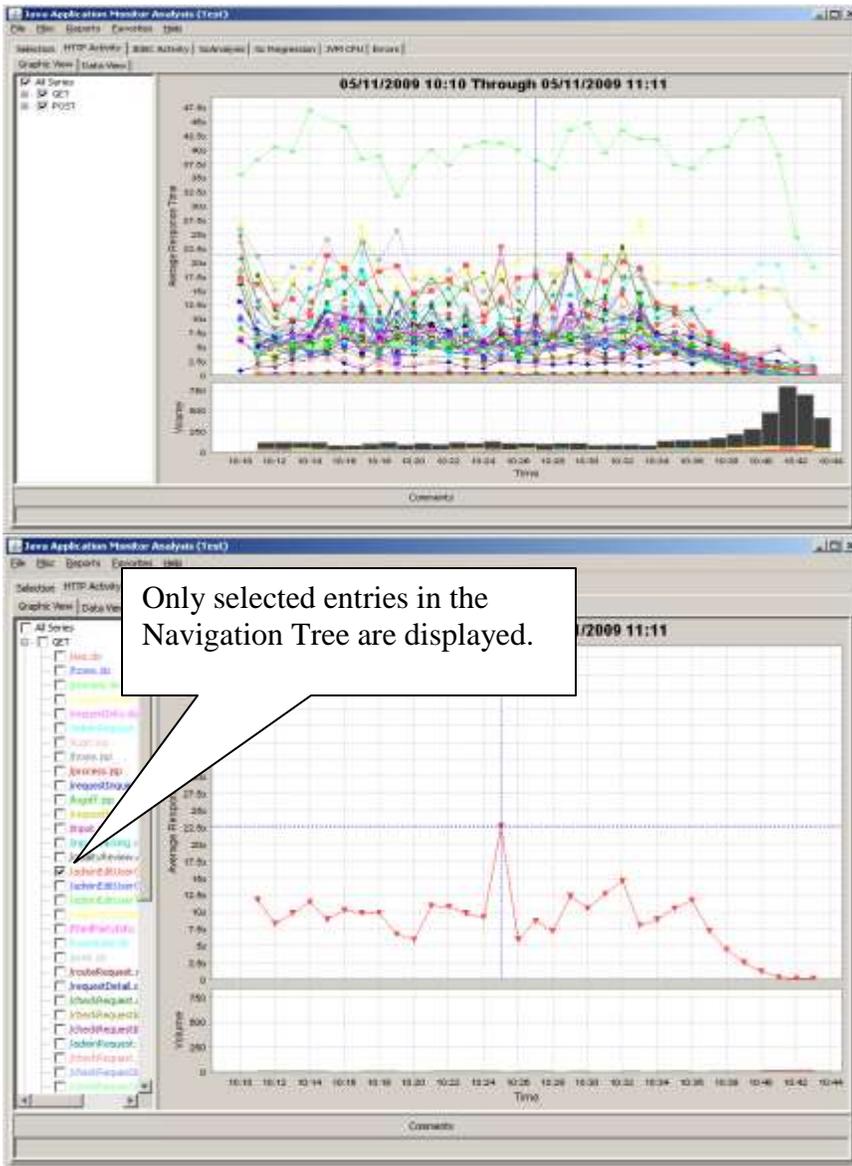
- Use the “group by” and the “group every” fields to control the granularity of the query (number of data points per series). You can group data by Seconds, Minute, Hour, Day or Month. The “group every” field lets you ‘tweak’ the “group by” field. For instance you may want to see data group in “15 minute” periods.
- You can add addition fields to your reports by enabling them in the “Fields” section.
- Use the “Field Order” dialog to control the ‘order’ of the fields. This let’s you control the “grouping” of data sets into logical section.



5. Navigation Tree

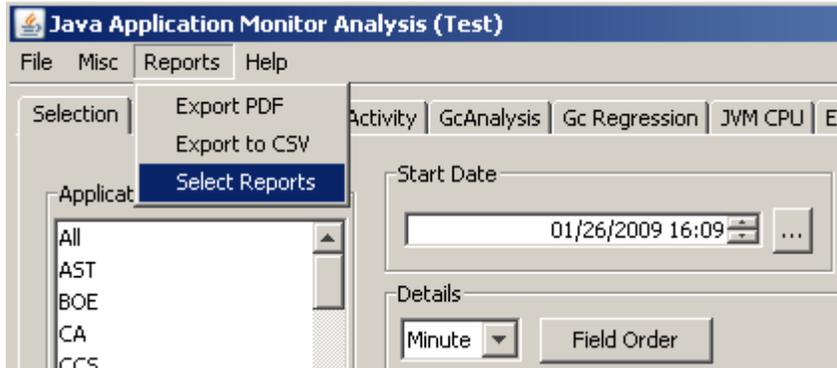
Once you have defined and executed a query, you have to be able to understand and interpret the data. The Analysis tool will help understand the data by providing but a graphic view and a data view.

To help overcome the “Information overload” due to the shear number of data elements, the analysis tool provides a “Navigation Tree” that allows you to “turn on” and “turn off” each data set or a group of data sets. The “Navigation Tree” also acts as a legend for the graphic view.



6. Export Data

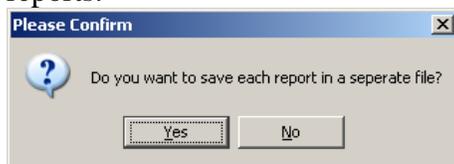
To export data, select the appropriate method from the “Reports” menu.



You can export data in two formats

- CSV (comma delimited); The CSV export is an export of the raw data used to define the views in the analysis tool. You can then do further analysis using any tool that supports a CSV import (Spread sheet, Load Runner Analysis tool).

When exporting data in CSV format, you can elect to export the data into multiple files (one for each report, this is recommended) or one file containing data for all reports.



Next select the location of the exported data (and file name if exporting to a single file).

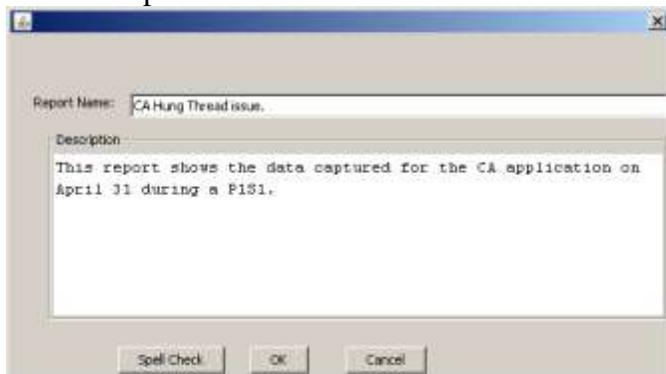


Select “Yes” to export the data with all dates specified as GMT. Otherwise, all dates are in the time zone of the server when the data was collected.

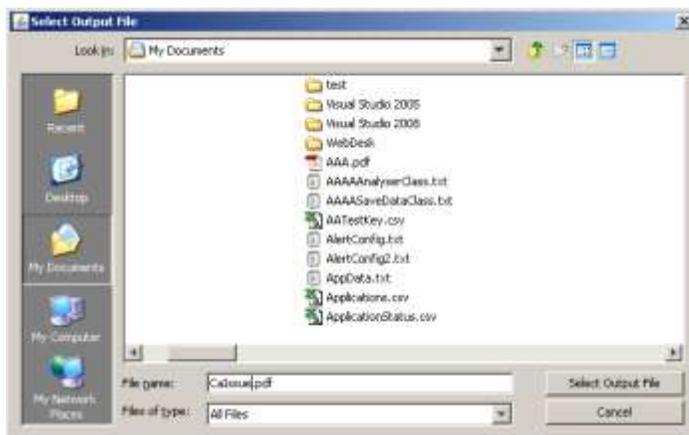


- PDF; The PDF export let's you create a portable version of the views provided by the analysis tool.

Assign the report and description. This information will appear in the summary page of the PDF report.

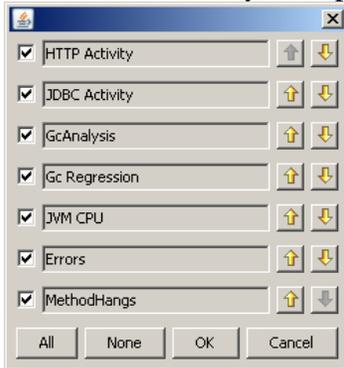


Next, select the desired file location and name



To avoid confusion, you should use always use .pdf or .csv as the file extension.

You may choose to export all reports or one or two select reports. You may also modify the order in which your reports will display in the output.



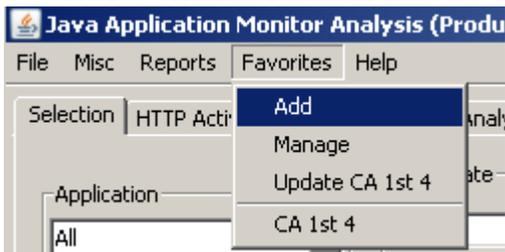
Select “OK” to complete the export.

7. Favorites

Once you have configured your select, you can save it as a “Favorite”. You can use Favorites menu to quickly recall frequently used queries.

For example, let’s assume a dependency between two application, “application #1 on server #1 calls application #2 on server#2”. You may want to view data from both applications at the same time. Once you defines the selection you can save that as a Favorite and easily recall it in the future.

Use the Favorites->Add menu to save the current selection.



You can use the Favorites ->Update ... menu option to save any changes to the current Favorite.

The Favorites -> Manage menu option allows you to;

- Create new folders
- Rearrange existing folders and favorites
- Rename folders and favorites (Select the entry and press F2)

8. Thread Analysis

Using the Thread Analysis panel you can view the execution code path for all instrumented classes and methods.

Use the Misc -> Thread Analysis menu option to open the Thread Analysis panel.

Due to the large amount of raw data available in Thread Analysis, there are a few restrictions. Before you can open the Thread Analysis panel, you must select a single application and a single server.

Set the start and end date / time to a short interval and select “Execute Query”.

The screenshot displays the Thread Analysis tool interface. At the top, there are input fields for Start Date (05/12/2009 07:41), End Date (05/12/2009 07:42), Application (CA), and Server (FE59024v). A "Execute Query" button is visible. Below these fields is a list of method calls, including:

- com.chase.ca.infrastructure.ui.servlet.RequestServlet.GET (3)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar.GET (2)
- _ibnisp.jsp.customer._customer_SP_summary.GET (14)
- com.chase.ca.infrastructure.ui.servlet.RequestServlet.GET (720)
- _ibnisp.jsp.customer._customer_SP_summary.GET (28)
- com.chase.ca.infrastructure.ui.servlet.RequestServlet.GET (187)
- _ibnisp.jsp.customer._customer_SP_search.GET (101)
- _ibnisp.jsp.customer._customer_SP_search.GET (11)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar.GET (2)
- com.chase.ca.infrastructure.ui.servlet.RequestServlet.GET (130)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar.GET (1-4)
- _ibnisp.jsp.infrastructure._header.GET (17)
- com.chase.ca.infrastructure.ui.servlet.ExtendedRequestServlet.POST (29)
- _ibnisp.jsp.product.servicing._card_SP_status_SP_change.GET (29)
- _ibnisp.jsp.customer._customer_SP_header.GET (1)
- _ibnisp.jsp.customer.support._error_SP_message_SP_header.GET (0)
- _ibnisp.jsp.product.servicing.support._ats_SP_transaction_SP_section.GET (0)
- _ibnisp.jsp.product.servicing.support._card_SP_status_SP_change_SP_details.GET (8)
- com.ibm.ws.adapter.jdbc.WS3dbDataSource.getConnection (0)
- com.ibm.ws.adapter.jdbc.WS3dbStatement.executeQuery (2)
- com.ibm.ws.adapter.jdbc.WS3dbDataSource.getConnection (1)
- com.ibm.ws.adapter.jdbc.WS3dbStatement.executeQuery (1)
- _ibnisp.jsp.product.support._legacy_SP_bill_SP_card_SP_delivery_SP_info.GET (18)
- com.ibm.ws.adapter.jdbc.WS3dbDataSource.getConnection (0)
- com.ibm.ws.adapter.jdbc.WS3dbStatement.executeQuery (2)
- com.ibm.ws.adapter.jdbc.WS3dbDataSource.getConnection (0)
- com.ibm.ws.adapter.jdbc.WS3dbStatement.executeQuery (1)
- com.ibm.ws.adapter.jdbc.WS3dbDataSource.getConnection (0)
- com.ibm.ws.adapter.jdbc.WS3dbStatement.executeQuery (2)
- com.ibm.ws.adapter.jdbc.WS3dbDataSource.getConnection (0)
- com.ibm.ws.adapter.jdbc.WS3dbStatement.executeQuery (1)
- _ibnisp.jsp.customer._customer_SP_search.GET (11)
- _ibnisp.jsp.customer._customer_SP_header.GET (2)
- _ibnisp.jsp.customer.support._error_SP_message_SP_header.GET (0)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar.GET (0)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar_SP_include.GET (0)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar_SP_incl.GET (3)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar.GET (1)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar_SP_include.GET (0)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar.GET (3)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar_SP_include.GET (1)
- com.chase.ca.infrastructure.ui.servlet.RequestServlet.POST (12)
- _ibnisp.jsp.infrastructure.navigation._standard_SP_index.POST (11)
- _ibnisp.jsp.infrastructure.navigation._navigation_SP_bar_SP_incl.POST (10)

A callout box labeled "Execution order" points to the list of method calls. On the right side of the interface, there are sections for "Messages", "CPU" (with Start and End fields), "Heap Percent" (with Start and End fields), and "Heap Usage" (with Start, End, and Consumed fields). At the bottom, there are fields for "End Time" and "End Date".

The execution tree lists method calls in chronological and execution order showing which methods are called by other methods.

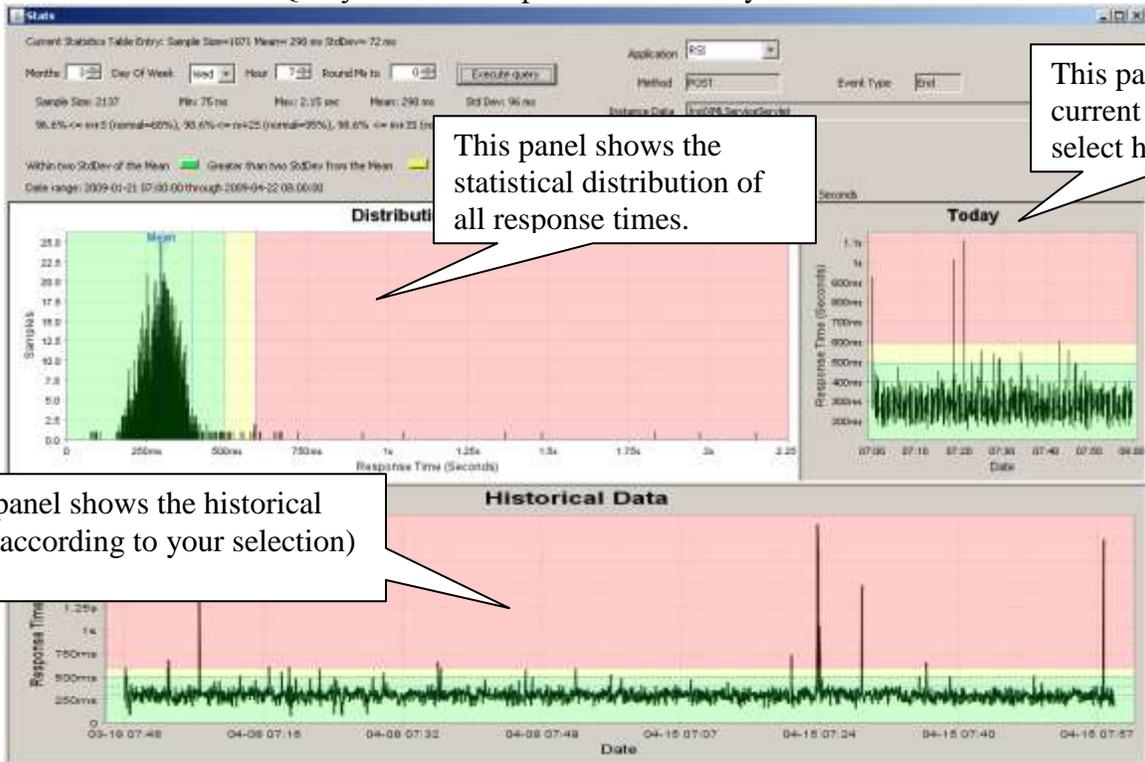
9. Statistics

When reviewing data you may see a data series that looks high. The question is, “is this normal”? You can use the Statistics view to find out. To go to the Statistics view, click on a data point in the graphic view and select “Statistics” from the menu.



The Statistics view is used to plot historical data for a given hour and day of the week into a distribution by response time. It can be helpful to understand what is ‘normal’ for a given data series. It also helps to understand the “shape” of the distribution. Most distributions are significantly skewed to the right.

Then select “execute Query”. Please be patient as this may take a minute or two.



10. Available Reports

Http Activity

The Http Activity report includes all instrumented methods of the instrumented Servlet classes. The methods include service, init and destroy. For the 'service' method (used to process an HTTP request) the report shows the HTTP method (GET / POST, ...) the URL (excluding the query string).

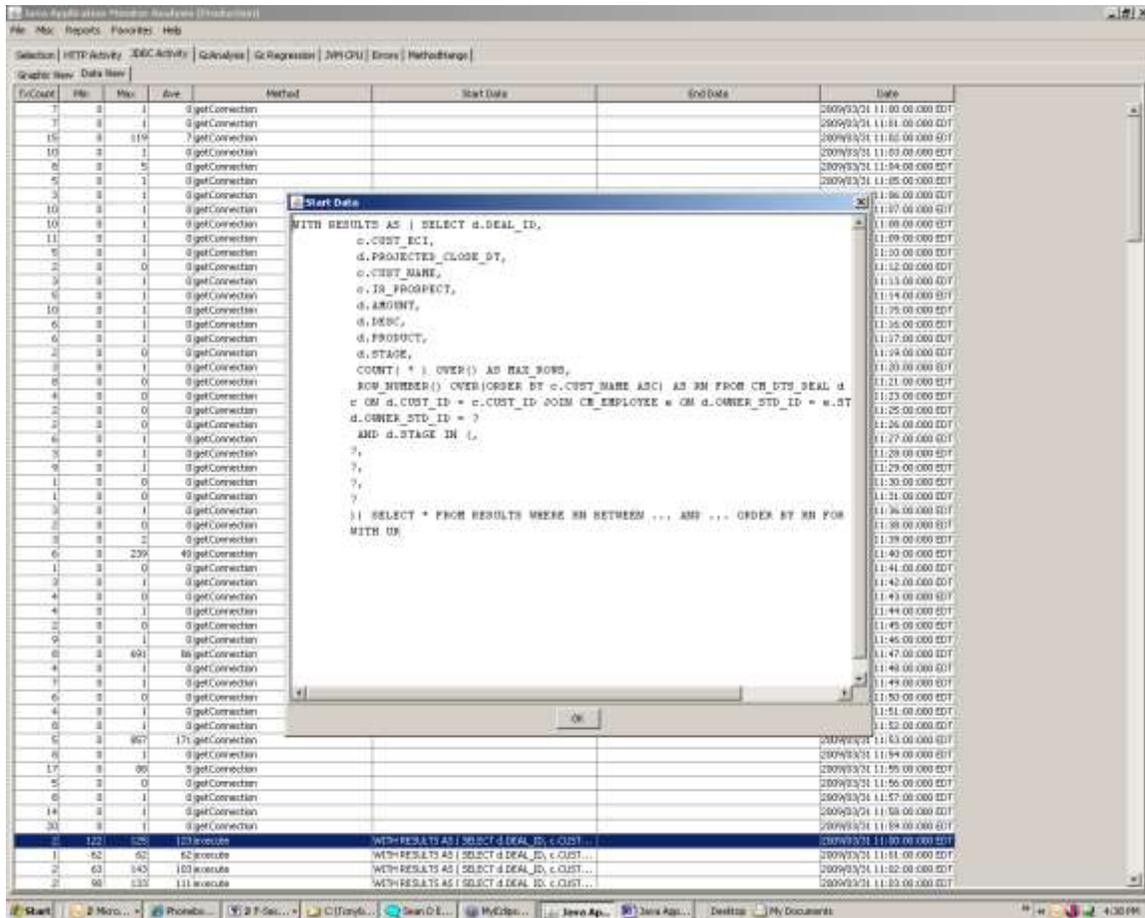


The screenshot shows the Java Application Monitor Analysis (Production) interface. The 'Selection' menu is set to 'HTTP Activity'. The 'Data View' tab is active, displaying a table with the following data:

ToCount	Min	Max	Ave	Method	Start Data	End Data	Date
2	1,731	1,798	1,764	GET	/cn/login.do		2009/03/31 10:59:00.000 EDT
191	2	6,277	1,760	GET	/cn/login.do		2009/03/31 11:00:00.000 EDT
306	1	30,143	2,960	GET	/cn/login.do		2009/03/31 11:01:00.000 EDT

Jdbc Activity

The Jdbc Activity report includes all instrumented methods of instrumented classes implementing the JDBC package. This report includes the SQL string, if applicable. When the SQL string is collected, it is obfuscated to prevent the collection of sensitive information.

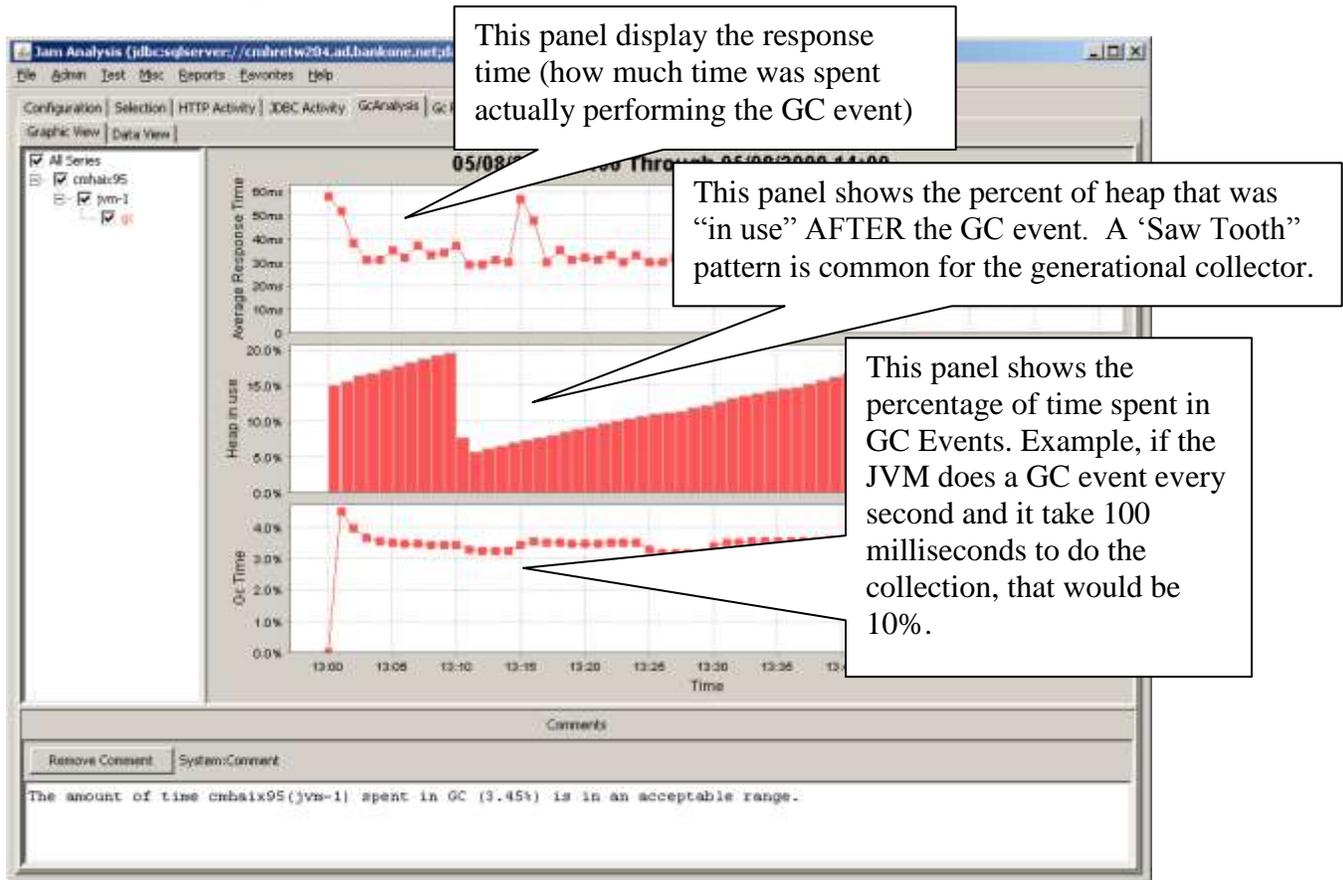


The screenshot shows the Java Application Monitor Analysis (Production) interface with the 'JDBC Activity' report selected. A 'Start Data' window is open, displaying the following SQL query:

```
WITH RESULTS AS ( SELECT d.DEAL_ID,
c.CUST_ID,
d.PROJECTED_CLOSE_DT,
c.CUST_NAME,
c.TR_PROSPECT,
d.AMOUNT,
d.DESC,
d.PRODUCT,
d.STAGE,
COUNT(*) OVER() AS RN FROM CH_DTS_DEAL d
c ON d.CUST_ID = c.CUST_ID JOIN CH_EMPLOYEE e ON d.OWNER_STD_ID = e.STD_ID AND d.STAGE IN (
?,
?,
?)
) SELECT * FROM RESULTS WHERE RN BETWEEN ... AND ... ORDER BY RN FOR
WITH UP
```

Gc Analysis

The Gc Analysis report includes data about the garbage collection.



GC time is a good indicator to the health of the JVM and the garbage collection system.

It is a ratio of the time spent doing garbage collection to the total processing time. For example, if the JVM performs a garbage collection once every second (1000 milliseconds) and it take 100 milliseconds to perform the collection, the result would be 10%.

For most applications,

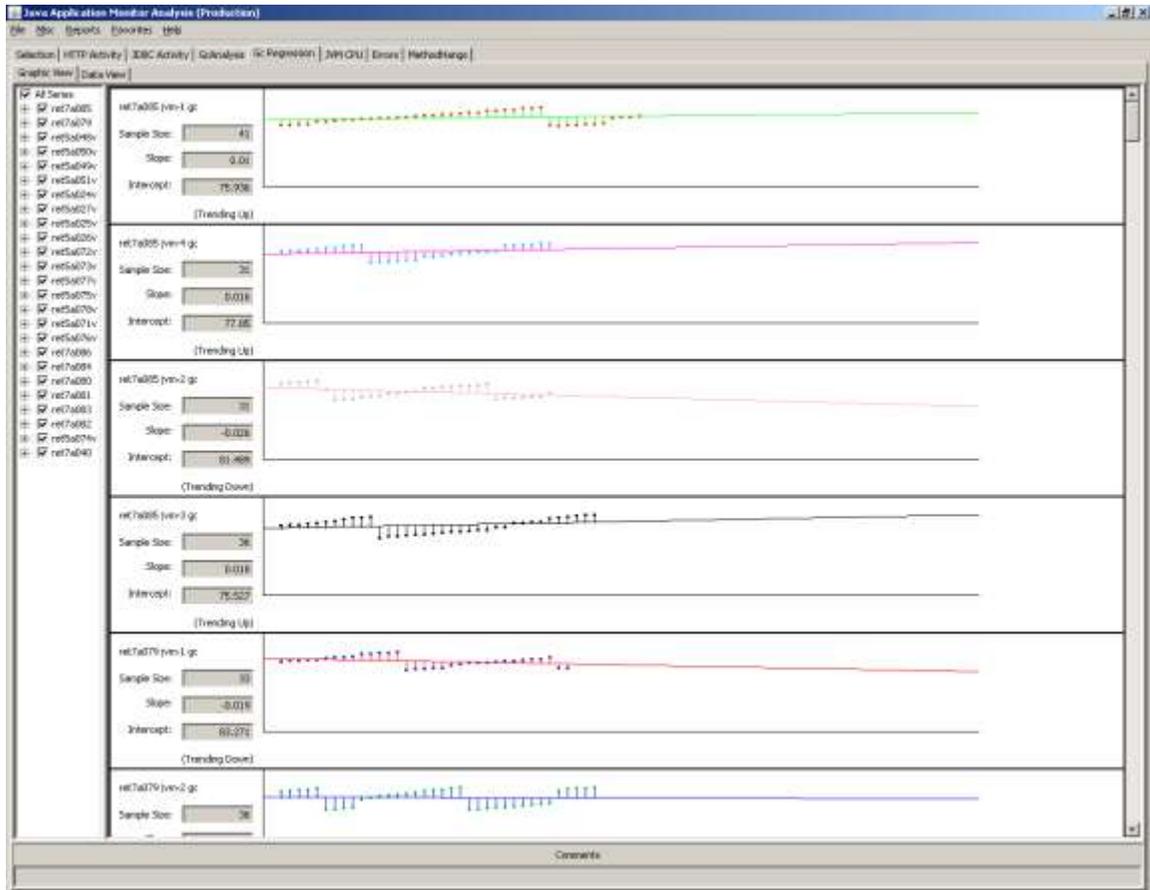
- < 7% is Green (healthy)

- 7% through 10 % is Yellow

- > 10% means the application response time is 'probably' degraded due to garbage collection.

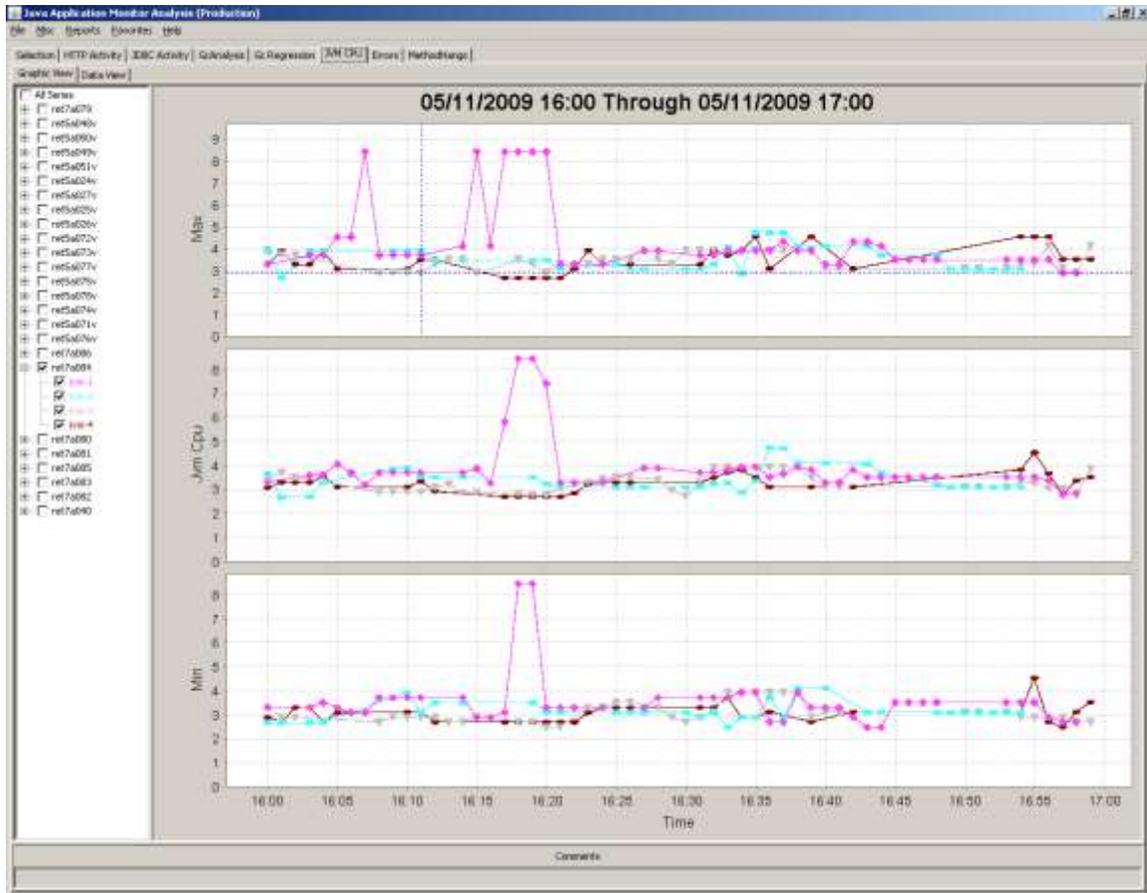
Gc Regression

The Gc Regression helps identify memory leaks in the Heap by projecting the heap usage into the future using a statistical linear regression curve. A positive slope is an indication that the application ‘may’ have a memory leak.



JVM CPU

The JVM CPU shows how much of the total system CPU was consumed by an individual JVM process. This helps us understand if the load is balanced between multiple JVMs and helps identify that some other process is competing with the application for resources.



Errors

Application errors, from instrumented methods, are shown in the Errors report. The report includes the Date and Time of the error, the Exception (type of error) and the Stack Trace of the current thread.

The screenshot shows a software application window titled "Errors". The main area contains a table with the following columns: Start Date, End Date, Server, JMX, Method, Date, and Stack Trace. The table lists several error entries, with one entry highlighted in blue. A "Stack Trace" dialog box is open, displaying a detailed list of Java stack frames.

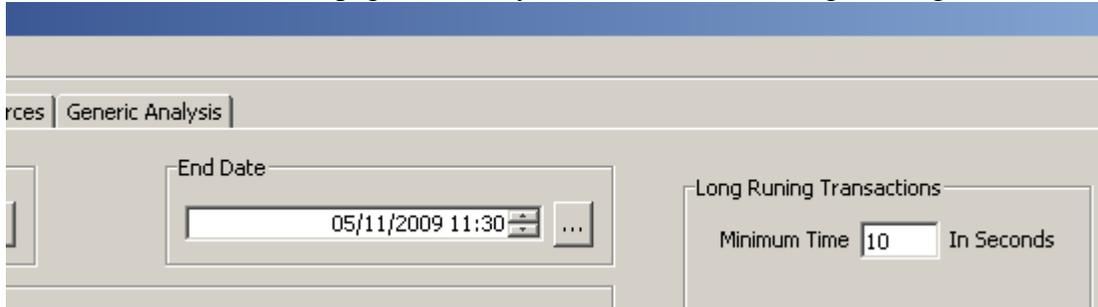
Start Date	End Date	Server	JMX	Method	Date	Stack Trace
2009/05/11 11:08:00 EDT	2009/05/11 11:08:00 EDT	com.ibm.ws...
2009/05/11 11:08:00 EDT	2009/05/11 11:08:00 EDT	com.ibm.ws...
2009/05/11 11:08:00 EDT	2009/05/11 11:08:00 EDT	com.ibm.ws...
2009/05/11 11:08:00 EDT	2009/05/11 11:08:00 EDT	com.ibm.ws...
2009/05/11 11:08:00 EDT	2009/05/11 11:08:00 EDT	com.ibm.ws...
2009/05/11 11:08:00 EDT	2009/05/11 11:08:00 EDT	com.ibm.ws...
2009/05/11 11:08:00 EDT	2009/05/11 11:08:00 EDT	com.ibm.ws...
2009/05/11 11:08:00 EDT	2009/05/11 11:08:00 EDT	com.ibm.ws...
2009/05/11 11:08:00 EDT	2009/05/11 11:08:00 EDT	com.ibm.ws...

```
com.ibm.ws.ut11.ThreadPoolWorker.run(ThreadPool.java:1473)
com.ibm.io.async.ResultHandler.runEventProcessingLoop(ResultHandler.java:40)
com.ibm.io.async.AsyncFuture.completed(AsyncFuture.java:136)
com.ibm.io.async.AbstractAsyncFuture.invokeCallback(AbstractAsyncFuture.java:146)
com.ibm.ws.tcp.channel.impl.AIOReadCompletionListener.futureCompleted(com.ibm.ws.tcp.channel.impl.AIOReadCompletionListener.java:104)
com.ibm.ws.tcp.channel.impl.HttpIOReadLink.complete(HttpIOReadLink.java:104)
com.ibm.ws.tcp.channel.impl.HttpIOReadLink.handleNewInformation(com.ibm.ws.tcp.channel.impl.HttpIOReadLink.java:104)
com.ibm.ws.webcontainer.channel.WCChannelLink.ready(WCChannelLink.java:104)
com.ibm.ws.webcontainer.WebContainer.handleRequest(WebContainer.java:704)
com.ibm.ws.jsp.webcontainerext.AbstractJSPExtensionServletWrapper.handleRequest(com.ibm.ws.webcontainer.servlet.GenericServletWrapper.handleRequest(com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrapper.java:104)
com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrapper.java:104)
com.ibm.ws.webcontainer.servlet.ServletWrapper.service(ServletWrapper.java:104)
com.ibm.ws.jsp.customer._customer_SF_search.service(ServletWrapper.java:104)
com.ibm.ws.jsp.customer._customer_SF_search.service(ServletWrapper.java:104)
```

Long Running Transactions

The Long Running Transactions report shows any transaction (or method call) during the specified time period that took longer than the user defined time to execute.

When you select the Long Running Transactions report, additional selection criterion is available on the selection page to allow you to define what a long running transaction is.

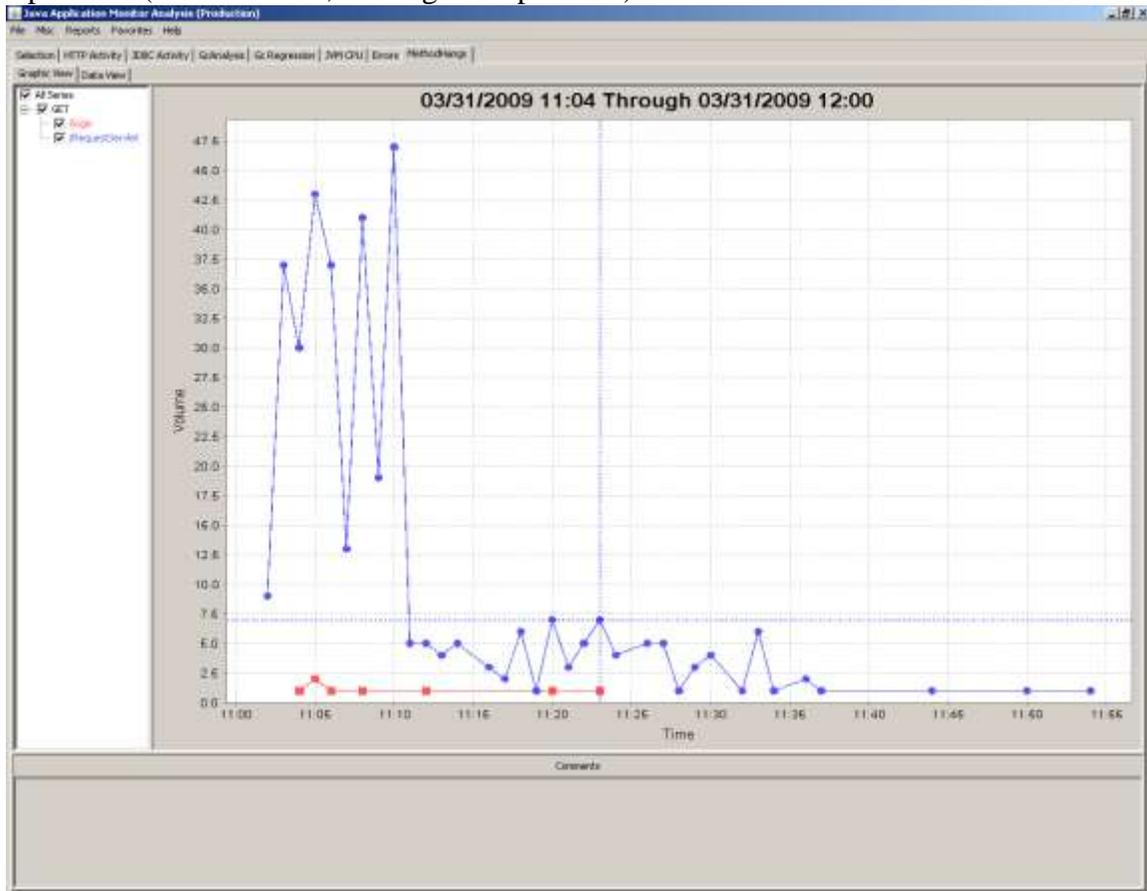


The screenshot shows a software interface with a blue header bar. Below the header, there are two tabs: "rces" and "Generic Analysis". The "Generic Analysis" tab is selected. In the center, there is a date and time selection field labeled "End Date" containing the text "05/11/2009 11:30" and a small calendar icon. To the right of this field is a "Long Running Transactions" section with a "Minimum Time" input field containing the value "10" and the text "In Seconds".

Method Hangs

The Method Hangs report shows any method that is considered “hung”. A “hung” method is one that started during the selected period but has not, as yet, ended.

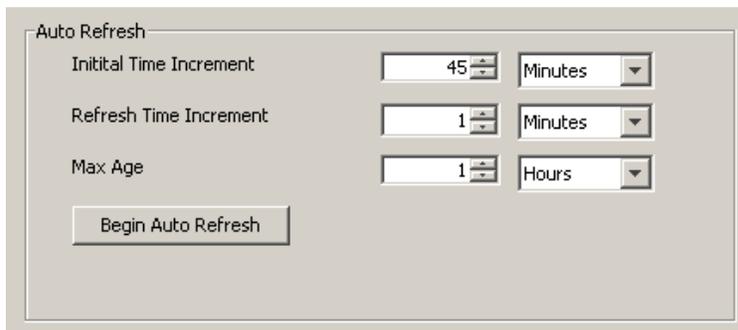
Our expectation is that this report is normally empty and any data here is an indication of a problem (the more data, the larger the problem).



11. Auto Refresh

In “Auto Refresh” mode, your reports will update automatically adding current data.

To begin Auto Refresh mode, select “Begin Auto Refresh” button from the Selection page, rather than the “Execute Query” button.



Auto Refresh

Initial Time Increment: 45 Minutes

Refresh Time Increment: 1 Minutes

Max Age: 1 Hours

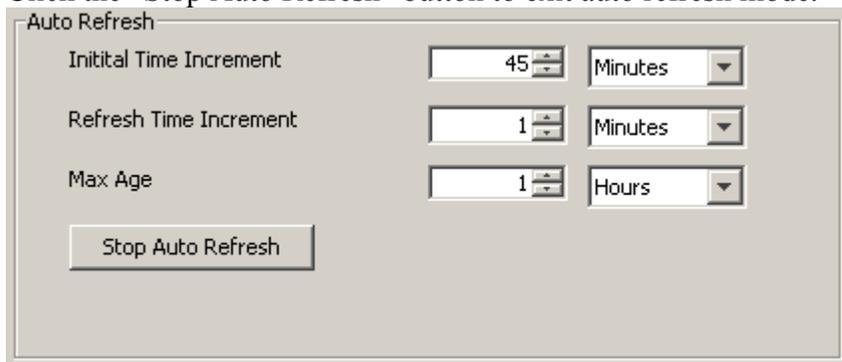
Begin Auto Refresh

- Initial Time Increment; when you begin Auto Refresh mode, all reports are populated with data starting with the selected “Start Date” and adding the initial time increment, up to the current time.
- Refresh Time Increment; this determines the frequency of the auto refresh.
- Max Age; you may run auto refresh mode for an extended period of time. As current data is added to the reports, anything older than then max age is removed.

While in auto refresh mode, you clear all reports using the File -> Data -> Clear menu option.



Click the “Stop Auto Refresh” button to exit auto refresh mode.



Auto Refresh

Initial Time Increment: 45 Minutes

Refresh Time Increment: 1 Minutes

Max Age: 1 Hours

Stop Auto Refresh

12. Data Cache (help menu)

The analysis tool uses several data caches to improve performance. If you encounter an “Out Of Memory” error, you may be able to continue by clearing one or more of the data caches.

To view the current cache information use the Help -> About menu option.

