Examining Precise for J2EE performance over time

This section includes the following topics:

- About the Activity workspace
- How the Activity workspace is structured
- About the Activity workspace analysis tabs
- How the identify performance problems in the Tree View

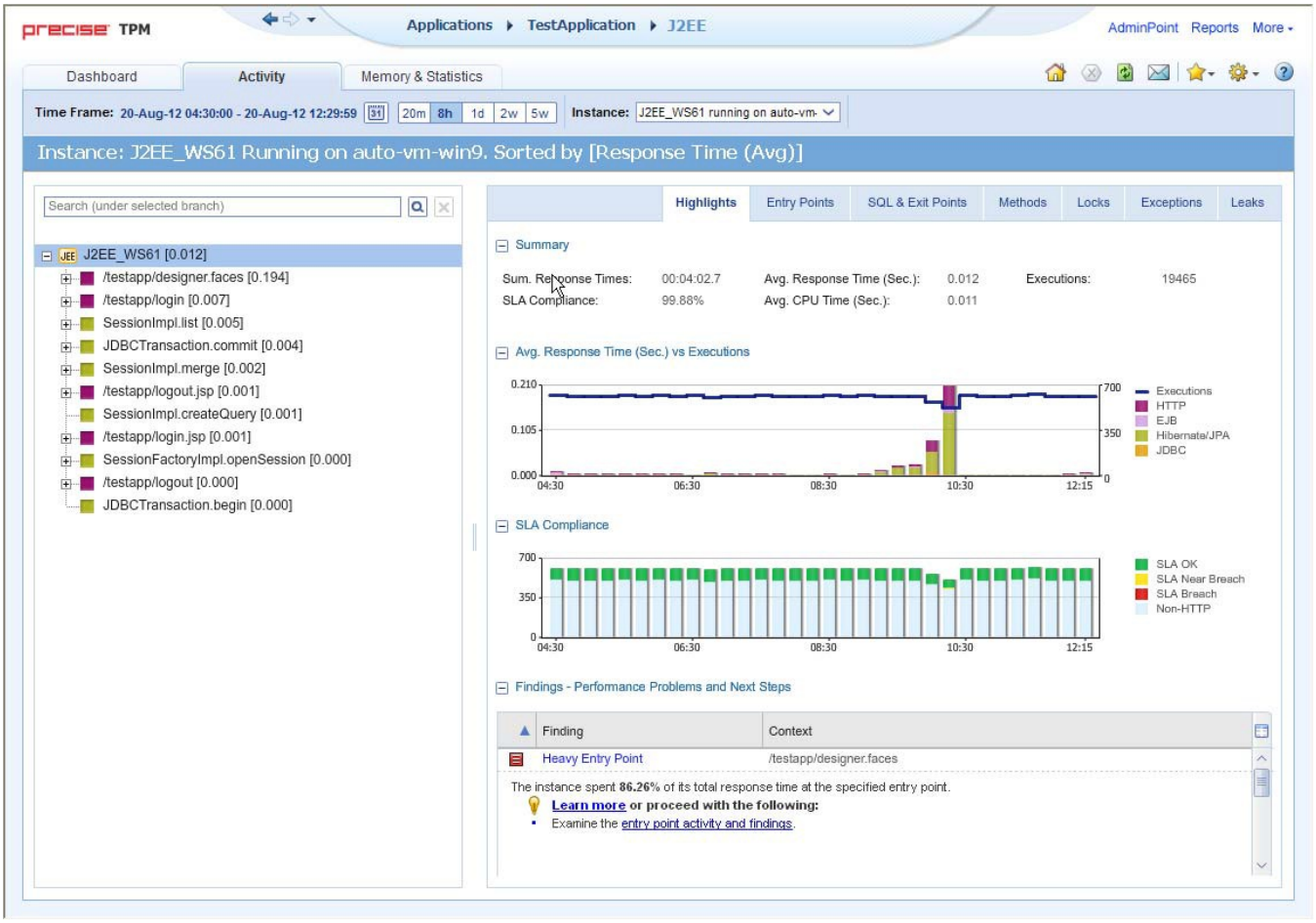
About the Activity workspace

The Precise for J2EE Activity workspace displays an execution tree of all monitored instances and various analysis tabs, enabling comprehensive and effective drilling down in the monitored instances and their invoked methods to locate specific performance issues and their underlying causes.

How the Activity workspace is structured

The Activity workspace is divided up into two main areas. The Main area (left pane) features an execution tree of all monitored instances. The Association area (right pane) features analysis tabs that provide specific information about the node selected in the execution tree in the left pane. The workspace heading displays the instance name, the server name, and what field the tree is sorted by.

Figure 1 Sample Activity Workspace



About the Main area

The Main area displays an execution tree of the monitored instance, broken down into the following node types:

- All instances (default)
- Instance
- URI/iView
- Method (all types)
- SQL (other exit points are methods)

Selecting a node displays information in context for that node and it's underlying call path in the analysis tabs in the Association area. See "About the Association area" on page 25.

Viewing information in the Main area

By default, the execution tree in the Main area is launched displaying the top instance level (either a selected instance or all monitored instances) and the entry point level only. Navigate through the execution tree by clicking the '+' sign next to the node name. This displays deeper layers of the tree, enabling you to view invoked methods and SQLs while maintaining the execution hierarchy.

> The degree to which you can drill down in an execution tree is defined by the filtering settings for your application. For more information, see Configuring Precise for J2EE in the Precise Administration Guide.

The following table describes the information displayed in the execution tree.

Table 1 Information displayed in the execution tree

Field	Description
Icon	Displays an icon reflecting the node type.
Name	Displays the name of the selected node. Depending on the tree view settings selected, the name will be displayed either its short name, or its full execution context.
Selected "Sorted by" value	Displays an aggregated value of the selected "sort by" option for the node's entire underlying call tree. For example, the JDBC contribution to JVM response time will be an aggregation of all JDBC methods times of that JVM.

To change the current tree view settings, see "Configuring Tree View Settings" on page 47.

Hovering over a node displays a ToolTip. The following table describes the information displayed in the ToolTip:

Table 2 Information displayed in a node's ToolTip

Field	Description
Type	Displays the node's method type.
Context	Displays the full execution context of the selected node.
Selected "Sorted by" type and value	Displays the field by which the tree view is sorted by and the value for the selected node and its underlying call tree.

The table below describes the "sort by" options available for the execution tree.

Table 3 Sort options for the execution tree

Sorted by	Description
Response Time (Avg)	Displays the average response time for each node and its underlying call path. The average response time is displayed in seconds.
Response Time (Sum)	Displays the total response time for each node and its underlying call path. The summed response time is displayed in hours: minutes: seconds: milliseconds.
Work Time (Avg)	Displays the average work time in seconds for each node. The average work time is displayed in seconds.
Work Time (Sum)	Displays the summed work time in seconds for each node. The summed work time is displayed in hours: minutes: seconds: milliseconds.
Executions	Displays the number of executions within each node.

All of the sort options can be displayed in ascending or descending order. You can also select the maximum number of executions to display under each node, the maximum number of results to display, and whether to display the method's long or short name.

See "Configuring Tree View Settings" on page 47.

When you right-click on an node in the execution tree, a popup menu is displayed. The following table describes the right-click menu options.

Searching within the execution tree

Above the execution tree in the Main area there is a search bar.

To search for a specific node

1. Enter the desired value in the search bar.
2. Click the search icon to the right of the search field. The heaviest results for the search value are emphasized in bold within the execution tree and their relevant call paths are opened.

> If the 'Show short name only' option is selected in the tree view settings dialog and the user enters the full name in the search value, the search will not return any results.

About the Association area

The Association area displays a variety of analysis tabs, corresponding to the selected node in the execution tree. See "About the Activity workspace analysis tabs" on page 25.

About the Activity workspace analysis tabs

The Activity workspace analysis tabs display indepth information about and in context of the node selected in the execution tree, shown within the tree in bold for referencing. While navigating through the analysis tabs, you can select different hyperlinked entities associated with the originally selected node. Selecting a hyperlinked node in an analysis tab will select that node within the execution tree, and all information displayed in the analysis tabs will be refreshed to correspond to the newly selected node.

The following table describes the analysis tabs available in the Association area of the Activity workspace and the node types for which each tab is displayed. As you navigate through the execution tree, the available analysis tabs change according to the selected node.

Table 4 Analysis tabs available in the Precise for J2EE Activity workspace

Tab Name	Description	Displayed for	For more information
Highlights	Displays a comprehensive performance overview of the node selected in the execution tree. This view reveals performance problems of the entire call tree invoked under the selected node.	All entities.	See "The Highlights tab" on page 26.
Load Balance	Displays information for the selected node throughout the monitored application.	All entities. This tab is only displayed when "All Instances" is selected.	See "About the Load Balance tab" on page 28.
Impact	Displays information regarding the selected node's impact on the application. It displays general performance data for the selected node in the selected time frame and specific information regarding its contribution to all entry points and methods by which it is directly invoked. This view provides the user with a comprehensive overview of the specific method's performance throughout the application.	• Methods • SQLs • Exit Points	See "About the Impact tab" on page 28.
Entry Points	Displays and compares performance data of the entry points in the selected time frame. This view enables efficient identification of problematic entry points.	Instances/All instances	See "About the Entry Points tab" on page 30.
SQL & Exit Points	Displays and compares performance data of the SQL and exit points in the selected time frame. This view enables efficient identification of problematic SQLs and exit points.	All entities except SQLs.	See "About the SQL & Exit Points tab" on page 31.
Methods	Displays information regarding all the methods and URI/iView invoked in the call tree under the selected node. The view provides a deeper look into a problematic node by showing comprehensive data for all invoked methods.	All entities except SQLs.	See "About the Methods tab" on page 32.
Locks	Displays all methods in the call tree under the selected node that were locked at any point. This view provides an overview of all locked methods and enables the user to see locking trends.	All entities except SQLs.	See "About the Locks tab" on page 33.
Exceptions	Displays data for exceptions thrown from the call tree under the selected node. This view pinpoints methods with a high exceptions rate.	All entities except SQLs.	See "About the Exceptions tab" on page 34.

Leaks	Displays the object allocation data for the call tree under the selected node.	Instances/All instances	See “About the Leaks tab” on page 35.

The Highlights tab

The Highlights tab appears when any node type within the execution tree is selected, and displays a comprehensive performance overview of the selected node. The highlights tab provides information regarding performance trends for the selection’s underlying call path and displays performance findings to facilitate focused navigation to the root cause of a performance issue.

The Highlights tab includes the following components:

- Summary area
- Avg. Response Time (Sec) vs Executions overtime graph
- SLA Compliance overtime breakdown graph
- Avg. JDBC Time (Sec) overtime graph (for SQL only)
- SQL Text (for SQL only)
- Destinations table (for SQL only)
- Findings table

Summary area

The summary area displays the following information for the selected node. The value displayed is the aggregated value for the selected node’s underlying call path (unless otherwise stated) in the selected time frame:

- **SLA Compliance**. Displays the relative percentage of HTTP entry point service requests that approached or breached the defined SLA thresholds.
- **Avg. Response Time (Sec)**. Displays the average response time for the selected node. The average response time is displayed in seconds.
- **Executions**. Displays the number of executions within the selected node.
- **Sum. Response Times**. Displays the total response time for the selected node. The summed response time is displayed in hours: minutes: seconds: milliseconds.
- **Avg. CPU Time (Sec)**. Displays the average time that CPU was consumed for the selected node and its underlying call path.
- **Method Type (when a method is selected only)**. Displays the method type.
- **Impact on Entry Point (when a method or an SQL is selected only)**. Displays the percent value of the selected node’s summed response time out of its entry point’s summed response time.
- **Max. Response Time (Sec) (when an SQL is selected only)**. Displays the peak response time detected in the selected time frame.

Avg. Response Time (Sec) vs Executions overtime graph

The Avg. Response Time (Sec) vs Executions overtime graph displays a bar for each time slice in the selected time frame that the selected node (or part of its underlying call path) was active. These bars display a breakdown of the invoked method types according to each method type’s average response time. The average response time for the entire time slice is compared to the number of executions in the time slice, displayed in linear format.

To the right of the table there is a legend detailing the method types displayed in the graph. Hovering over any point in the graph displays a ToolTip detailing the date and time of the selected time slice bar, the number of executions, and the average response times per method (in seconds) for that time slice.

SLA Compliance overtime breakdown graph

The SLA Compliance overtime graph displays a bar for each time slice in the selected time frame that the selected node (or part of its underlying call path) was active. These bars display a breakdown of the SLA compliance of the invoked methods as follows:

> SLA compliance is calculated for HTTP entry points only.

- **Green**. The invoked method’s SLA is below the defined SLA threshold.
- **Yellow**. The invoked method’s SLA is approaching the defined SLA threshold.
- **Red**. The invoked method’s SLA exceeded the defined SLA threshold.
- **Blue**. Top level methods that do not have SLA thresholds defined for them (for example, EJB, JDBC, and so on).

These executions will appear as “Non-HTTP” executions.

> Information for non-HTTP methods will not appear in the legend and in the graphs, but will appear in the ToolTip.

Avg. JDBC Time (Sec) overtime graph (for SQL only)

The Avg. JDBC Time overtime graph displays a bar for each time slice in the time frame that the SQL was active. The average JDBC time (comprised of the JDBC time only as it is the lowest point in the execution tree and has no underlying branches) is compared to the number of executions in the time slice, displayed in linear format.

SQL Text (for SQL only)

This area displays the entire name of the selected SQL.

Destinations table (for SQL only)

Displays a table at all SQL destinations with details such as:

- DBMS
- DB Server
- DB Name
- Connection string details

Findings table

The Findings table shows findings for the selected entity and its underlying call tree. Context names that are too long to display are automatically shortened using ellipses.

For more information regarding findings, see “How to identify performance problems” on page 49

About the Load Balance tab

The Load Balance tab displays the selected entity and compares its performance in all the contexts it was invoked by in the selected time frame. If “All instances” is selected in the execution tree, it will show the instances table (similar to the view provided in the Dashboard workspace).

> This tab is only displayed when “All Instances” is selected as the top level of the execution tree. When a specific instance is selected as the top level of the execution tree, this tab is not displayed.

The Load Balance tab includes the following:

- Load Balancing between JVMs table
- Avg. Response Time (Sec) overtime graph
- Executions overtime graph

Load Balancing between JVMs table

The Load Balancing between JVMs table displays the following information for all instances:

- **Instance**. Displays the instance name. If you select the highlighted instance, the execution tree will update itself to display the specific instance’s execution tree.

  > Once a specific instance is selected, the Load Balance tab will no longer appear.

- **Server**. Displays name of the server on which the instance is running.
- **Avg. Response Time (Sec)**. Displays the average response time for the selected node. The average response time is displayed in seconds.
- **Executions**. Displays the number of executions within the selected node.
- **Sum. Response Times**. Displays the total response time for the selected node. The summed response time is displayed in hours: minutes: seconds: milliseconds.

Avg. Response Time (Sec) overtime graph

The Response Time (Avg) overtime graph displays and compares the average response time of the selected instance in the selected time frame against the average response time frame for the AppTier. If the average response time is identical, the overtime graph will display information for the selected JVM on top of the AppTier average.

Executions overtime graph

The Executions overtime graph displays and compares the average number of executions for the selected instance in the selected time frame against the average number of executions for the AppTier. If the average number of executions is identical, the overtime graph will display information for the selected JVM.

About the Impact tab

The Impact tab displays information regarding the selected method’s impact on the application. It displays general performance data in context for the selected node and specific information regarding its contribution to all entry points and methods in the application by which it is directly invoked. This provides the user with a comprehensive overview of the specific method’s performance throughout the application.

The Impact tab displays the following information:

- Summary area
- Impact on All Entry Points table
- Impact on All Direct Callers table

Summary area

The summary area displays the following information for the selected entity. The value displayed is the aggregated value for the selected entity's underlying call tree in the selected time frame:

- **Method (only when a method is displayed, not SQL)**. Displays the method type.
- **Avg. Response Time (Sec)**. Displays the average response time for the selected node. The average response time is displayed in seconds.
- **Executions**. Displays the number of executions within the selected node.
- **Sum. Response Times**. Displays the total response time for the selected node. The summed response time is displayed in hours: minutes: seconds: milliseconds.
- **Max. Response Time (Sec)**. Displays the peak response time detected in the selected time frame.
- **Min. Response Time (Sec)**. Displays the lowest response time detected in the selected time frame.
- **Avg. Work Time (Sec) (not displayed for SQLs)**. Displays the average actual work time of the selected node without the underlying call path.
- **Max. Work Time (Sec) (not displayed for SQLs)**. Displays the maximum work time.
- **Avg. CPU Time (Sec)**. Displays the average time that CPU was consumed for the selected node and its underlying call path. (Not displayed for SQLs).
- **Max. CPU Time (Sec) (only when a method is displayed, not SQL)**. Displays the peak CPU time detected for the selected node.

Impact on All Entry Points table

The Impact on All Entry Points table displays the following information for every entry point that the selected method was invoked by in the selected time frame:

- **Entry Point**. Displays the name of the entry point from which the entity was invoked.

  > Selecting the hyperlinked entry point name selects the entry point in the execution tree and displays the highlights tab for the entry point.

- **Work Impact on Entry Point**. Displays the percent value of the selected node's summed response time out of its entry point's summed response time.
- **Method Avg. Response Time (Sec)**. Displays the method's average response time.
- **Executions by Entry Point**. Displays the number of times the selected method was invoked by the specific entry point.
- **Sum. Method Response Times**. Displays the method's total response time.
- **Sum. Entry Point Response Times**. Displays the entry point's total response time.

Impact on All Direct Callers table

The Impact on All Direct Callers table displays the following information for every method that directly invoked the selected method:

- **Caller**. Displays the name of the method that directly invoked the selected method.

  > Selecting the hyperlinked direct caller name selects the method in the execution tree and displays the Highlights tabs for the method.

- **Work Impact on Caller**. Displays the percentage of the selected method's total response time out of the direct caller method's total response time.
- **Method Avg. Response Time (Sec)**. Displays the method's average response time.
- **Executions by Caller**. Displays the number of times the selected method was executed by the specific direct caller method.
- **Sum. Method Response Times**. Displays the method's total response time.
- **Sum. Caller Response Times**. Displays the direct caller method's total response time.

About the Entry Points tab

The Entry Points tab displays the top level URI/iView or methods (EJB or others) that are monitored for this instance, comparing performance data of the detected entry points within the monitored application in the selected time frame. This view enables efficient identification of problematic entry points and, in the event of many entry points from the same instance showing distress, problematic instances as well.

The Entry Points tab includes the following components:

- Entry Points table
- Avg. Response Time (Sec) vs. Executions overtime bar graph

> The SLA Compliance data can be switched to display as a percentage in the table, or in a ToolTip.

Entry Points table

The Entry Points table displays the following information for the top 30 entry points in the monitored instance:

- **Icon**. Displays the entry point type.
- **Name**. Displays the entry point name.

> Selecting the hyperlinked entry point name selects the entry point in the execution tree and displays the Highlights tab for the entry point.

- **Avg. Response Time (Sec)**. Displays the average response time for the selected node. The average response time is displayed in seconds.
- **Executions**. Displays the number of executions within the selected node.
- **SLA Compliance**. Displays the relative percentage of HTTP entry point service requests that approached or breached the defined SLA thresholds.
- **Type**. Displays the method type name.
- **Sum. Response Times**. Displays the total response time for the selected node. The summed response time is displayed in hours: minutes: seconds: milliseconds.
- **Avg. Work Time (Sec)**. Displays the average actual work time of the selected node without the underlying call path.
- **Sum. Work Times**. Displays the total work time for the selected node without the underlying call path.
- **Max. Response Time (Sec)**. Displays the peak response time detected in the selected time frame.
- **Min. Response Time (Sec)**. Displays the lowest response time detected in the selected time frame.
- **Avg. CPU Time (Sec)**. Displays the average time that CPU was consumed for the selected node and its underlying call path.
- **Sum. CPU Time (Sec)**. Displays the total CPU time.
- **Max. CPU Time (Sec)**. Displays the peak CPU time detected for the selected node.
- **CPU Work Impact %**. Displays the percent value of the selected node's CPU time out of its entry point's CPU time.
- **Avg. CPU Work Time (Sec)**. Displays the average actual work time that CPU was consumed for the selected node without its underlying call path.
- **Sum. CPU Work Times**. Displays the total CPU work time for the selected node without its underlying call path.

Avg. Response Time (Sec) vs. Executions overtime bar graph

The Avg. Response Time (Sec) vs. Executions overtime bar graph displays the relationship over time between the average response time of the entry point selected in the table above and the number of times the entry point was executed in the same time frame. Hovering over any point in the bar graph will display a ToolTip with the date and time of the time slice, the average response time, and the number of executions.

About the SQL & Exit Points tab

The SQL & Exit Points tab displays the following information for all SQLs and exit points in the call tree beneath the selected entity.

- Heaviest Exit Points Invoked Directly and Indirectly table
- Avg. Response Time (Sec) vs. Executions overtime bar graph
- Destinations table

The exit points tab can provide information for the following:

- SQL statements
- Exit points methods (each has its relevant type)
- Custom exit points. Custom exit points connect Precise for J2EE to technologies within your environment that Precise cannot directly monitor. Examples are the Documentum Content Server, the Mainframe, etc. Connecting to these technologies, Precise for J2EE can provide you with in-depth information regarding performance bottlenecks seen within your Precise application and pinpoint the root cause from within the external technology. By default, this feature is not configured in Precise for J2EE. For more information regarding enabling and configuring custom exit points, contact Customer Support.

Heaviest Exit Points Invoked Directly and Indirectly table

> When All Instances is the selected top level, this table will be called, Heaviest Exit Points by Work Time - ones that have at least 1% impact on the total.

The Heaviest Exit Points Invoked Directly and Indirectly table displays the following information for the selected entity:

- **Icon**. Indicates the method type or designated entity it represents.
- **Name**. Displays the exit point name. If the exit point is an SQL, the short SQL text is displayed. If the exit point is a method, the method's short name is displayed.
- **Work Impact**. Displays the percentage of work time of the selected exit point in relation to the total work time of the invoking method.
- **Avg. Response Time (Sec)**. Displays the average response time for the selected node. The average response time is displayed in seconds.
- **Executions**. Displays the number of executions within the selected node.
- **Type**. Displays the method type name.
- **Sum. Response Times**. Displays the total response time for the selected node. The summed response time is displayed in hours: minutes: seconds: milliseconds.
- **Max. Response Time (Sec)**. Displays the peak response time detected in the selected time frame.
- **Min. Response Time (Sec)**. Displays the lowest response time detected in the selected time frame.

Avg. Response Time (Sec) vs. Executions overtime bar graph

The Avg. Response Time (Sec) vs. Executions overtime bar graph displays the relationship over time between the average response time of the exit point selected in the table above and the number of times the exit point was executed in the same time frame. Hovering over any point in the bar graph will display a ToolTip with the date and time of the time slice, the average response time, and the number of executions.

Destinations table

The Destinations table displays all destinations for the selected entity:

- **Name**. Displays the destination name that can be either the Web service request name, 'DB Name' on 'Server Name' for SQLs, and so on.
- **Type**. Displays the exit point destination type such as, Web service, DB type, and so on.
- **Work Impact**. Displays the percentage of work time of the selected exit point in relation to the total work time of the invoking method
- **Executions**. Displays the number of executions within the selected node.

About the Methods tab

The All Methods tab contains both URI/iView and methods from the entire call tree beneath the current context - instance. All data is aggregated from the underlying call tree.

The Methods tab includes the following components:

- Methods Invoked Directly and Indirectly
- Avg. Response Time (Sec) vs. Executions for selected method overtime bar graph

Methods Invoked Directly and Indirectly

On top appears the comparison table called “All Methods Invoked Directly and Indirectly”.

> This will be Heaviest methods by work time when all instances in the top level in the executions table, with 1% impact min.

- **Icon**. Displays the entry point type.
- **Name**. Displays the entry point name.

  > Selecting the hyperlinked entry point name selects the entry point in the execution tree and displays the Highlights tab for the entry point.

- **Work Impact**. Displays the percentage of work time of the selected exit point in relation to the total work time of the invoking method
- **Avg. Response Time (Sec)**. Displays the average response time for the selected node. The average response time is displayed in seconds.
- **Executions**. Displays the number of executions within the selected node.
- **Type**. Displays the method type name.
- **Sum. Response Times**. Displays the total response time for the selected node. The summed response time is displayed in hours: minutes: seconds: milliseconds.Avg. Work Time (Sec) - displays the average actual work time of the selected node without its underlying call path.
- **Avg. Work Time (Sec)**. Displays the average actual work time of the selected node without the underlying call path.
- **Sum. Work Times**. Displays the total work time for the selected node without its underlying call path.
- **Max. Response Time (Sec)**. Displays the peak response time detected in the selected time frame.
- **Min. Response Time (Sec)**. Displays the lowest response time detected in the selected time frame.
- **Avg. CPU Time (Sec)**. Displays the average time that CPU was consumed for the selected node and its underlying call path.
- **Sum. CPU Times**. Displays the total CPU time.
- **Max. CPU Time (Sec)**. Displays the peak CPU time detected for the selected node.
- **CPU Work Impact %**. Displays the percent value of the selected node's CPU time out of its entry point's CPU time.
- **Avg. CPU Work Time (Sec)**. Displays the average actual work time that CPU was consumed for the selected node without its underling call path.
- **Sum. CPU Work Times**. Displays the total CPU work time for the selected node without its underlying call path.

Avg. Response Time (Sec) vs. Executions for selected method overtime bar graph

The Avg. Response Time (Sec) vs. Executions overtime bar graph displays the relationship over time between the average response time of the exit point selected in the table above and the number of times the exit point was executed in the selected time frame. Hovering at any point in the bar graph will display a ToolTip with the date and time of the time slice, the average response time, and the number of executions.

About the Locks tab

The Locks tab displays locked methods from the entire call tree beneath the selected node. All data is aggregated from the underlying call tree.

The Locks tab also displays the top level URI or iView, or methods (such as: URI, EJB, or others) that are monitored for this instance.

> The locking methods themselves are not displayed on the tree.

The “lock” identifier is calculated at runtime. It is based on the object name that is being locked (synchronized). It is possible to use the same object more than once during the same method. In that case, Precise for J2EE adds an index for each of the lock events, in order to enable the developer to distinguish between the two lock events.

For example:

```
Public void foo(){
     //run some code...
     Synchronized (myLockObject){// will be labeled as myLockObject+0
          //run some synchronized code...
     )
     //run some more code...
     // and then we need to have another synchronized block, locking the same object as before
     Synchronized (myLockObject){// will be labeled as my LockObject+1
          // run some more synchronized code...
     )
```

The Locks tab includes the following components:

- Locked Methods under the Selected Entity
- Methods Simultaneously Locked with the Method Selected Above
- Avg. Lock Time (Sec) of the Methods Selected in the Tables Above

Locked Methods under the Selected Entity

- **Icon**. Indicates the method type.
- **Name**. Displays the short name of the actual method that participated in a lock in the selected time frame. The long method name is shown in the ToolTip.

  > Clicking on a method name selects that method within the execution tree and refreshes the Activity workspace data for the selected method. The Activity workspace will open the Locks tab for the selected method.

- **Lock Impact**. Displays the relative percentage of the selected entity’s lock time, out of the instance’s summed response time.
- **Avg. Lock Time (Sec)**. Displays the average time the selected entity was locked.
- **Lock Acquisitions**. Displays the number of locks by the selected entity in the selected time frame.
- **Sum. Lock Times**. Displays the total time that the selected entity was locked in the selected time frame.

Methods Simultaneously Locked with the Method Selected Above

When a method is selected in the Locked Methods under the Selected Entity table, this table displays all actual methods (and not a method in their call path) that participated in a lock with the method selected above.

- **Icon**. Indicates the method type.
- **Name**. Displays the short name of the actual method that participated in a lock in the selected time frame. The long method name is shown in the ToolTip.

  > Clicking on a method name selects that method within the execution tree and refreshes the Activity workspace data for the selected method. The Activity workspace will open the Locks tab for the selected method.

- **Lock Impact**. Displays the relative percentage of the selected entity lock time, out of the instance’s summed response time.
- **Avg. Lock Time (Sec)**. Displays the average time the selected entity was locked.
- **Lock Acquisitions**. Displays the number of locks by the selected entity in the selected time frame.
- **Sum. Lock Times**. Displays the total time that the selected node was locked in the selected time frame.

Avg. Lock Time (Sec) of the Methods Selected in the Tables Above

The Avg. Lock Time (Sec) of the Methods Selected in the Tables Above overtime bar graph shows an overtime view of the periods where the methods selected in the tables above acquired locks. This view enables you to understand the mutual locking relationship of the two methods - are they always locked at the same times (meaning, they lock each other only), is one locking (indicated by a short lock time) and other locked (indicated by a long lock time), and so on. Hovering at any point in the bar graph will display a ToolTip with the date and time of the time slice, the average lock time, and the number of locks.

About the Exceptions tab

The Exceptions tab displays data for exceptions thrown from the entire call tree beneath the current context - instance. All data is aggregated from the underlying call tree.

The Exceptions tab includes the following components:

- Exceptions Thrown Directly and Indirectly
- Locations for Selected Exception
- Stack Traces

Exceptions Thrown Directly and Indirectly

On top appears a comparison table which summarizes the number of exceptions.

> Exceptions monitoring is turned off by default. To turn on exceptions monitoring, go to Settings>Monitor Settings. See “Configuring Monitor Settings” on page 41.

All columns can be sorted. The default sort is by Count column.

- **Thrown Class**. Displays the short name of the class. The long name, including packages, appears in the ToolTip.
- **Count**. The number of times the exceptions class was thrown.
- **Last Detected On**. Displays the exact date and time (to the second) that the thrown exceptions classes were detected.

Locations for Selected Exception

- **Location**. Displays the short location name of the exception. The long name appears in the ToolTip.
- **Count**. Displays the number of times the exception was thrown from the specific location.
- **Message**. Displays the message that was set in the thrown class.
- **Exception Cause**. Displays the exception cause (when available).
- **Monitored Caller**. Displays the direct calling method where the exception was thrown.

Stack Traces

On the bottom appears the title “Exception Stack Traces”. This table displays information where each row is a specific occurrence of the exception selected on top. There is one row per stack trace on this table.

Catches exceptions during a time slice, based on the instrumentation configuration (switched off by default) and filtering criteria.

See Precise Administration Guide, Appendix A, regarding configuring exception seeker options.

About the Leaks tab

The Leaks tab displays object allocation data from the entire call tree beneath the current context - instance. All data is aggregated from the underlying call tree.

> The Leaks tab is only relevant for instance or All Instances levels.

The Leaks tab includes the following components:

- Leak Candidates Allocated Directly and Indirectly
- Live Objects Count
- Stack Trace Details

Leak Candidates Allocated Directly and Indirectly

- **Allocation Method**. The method name where the object was allocated.
- **Object Type**. The object, collection, array, or string buffer. It contains values such as: “Integer”, “String”, or “MyObject”
- **Growth Rate**. The rate that the number of objects grows over time. This is displayed in percent for growth rate per minute.

Live Objects Count

On top appears an objects comparison table.

All columns can be sorted. The default sort is by Growth Rate (%) column.

> Leaks monitoring is turned off by default. To turn on Leaks monitoring, go to the Settings menu. See “Configuring Monitor Settings” on page 41.

The Leaks tab displays details about the top 64 allocation sites with the largest total live allocated objects in the selected time frame. Each allocating method displays the collection’s stack trace in its ToolTip. The stack trace displays the executions leading backwards from the allocation site to the application’s initial execution.

For the top *n* allocating methods, the chart shows the approximate number of live objects in the collection, or its growth rate (according to the sorting column). Negative trends can be identified using these charts by detecting memory leaking locations.

Aggregation in the table is according to: Allocating Method name, the Object Type, and the full path.

The table below describes the columns in the Leak Candidates Allocated Directly and Indirectly table.

Table 5 Columns in the JVM Leak Candidates Allocated Directly and Indirectly table

Item	Description
Chart Legend Color	Color icon showing the color representing the row in the graph.
Allocated Method	The method name where the object was allocated.
Object Type	The Object, Collection, Array, or String Buffer. It contains values such as: “Integer”, “String”, or “MyObject”.
Growth Rate (%)	Rate of growth (in percents) of the number of objects over time (objects/minutes). This column is default for sorting.
Live Objects Count	Number of live instances of an object. Live Objects Count can be displayed as a “waterfall” bar, or in numbers.

Stack Trace Details

Stack Trace details appear in the expanded view that appears when hovering over each row’s complete path (Stack Trace).

How to identify performance problems in the Tree View

You can identify a performance problem in Tree View by doing one or more of the following:

- Identifying slow Service Requests
- Identifying slow Methods
- Identifying the most invoked Service Request or Method
- Searching for a specific Service Request or Method

The performance attribute by which the Tree View is sorted is shown on the workspace heading as well as on the ToolTip displayed when you place the cursor over a tree node. The number in brackets is the value of the entity according to its sort attribute.

Identifying slow Service Requests

To identify slow Service Requests

1. In the Time Frame list, select the period of time you want to analyze.
2. In the Instance list, select the J2EE instance you want to analyze.
3. Sort the tree by the performance attribute you want to explore (for example, Response Time (Avg) or Response Time (Sum). The default is Response Time (Avg), but you can change it using the Tree View Settings.
4. In the Tree pane, select the Service Request you want to analyze. Focus on the Service Request that has a performance counter (for example, Response Time) that is higher than it should be.
5. In the Details pane, look at both the Response Time graph and the Executions graph. Select a problematic time period to investigate and hover over it with the mouse. Additionally, analyze the details that are displayed in the Overview.
6. On the displayed ToolTip, look for the time period.
7. To view the problem in more detail, in the Time Frame list, select a narrower time frame which includes the time period that is displayed on the ToolTip.
8. To investigate the root cause of the slow performance, expand the Service Request tree node to see its invoked methods. Continue with the Identifying slow Methods procedure.

Identifying slow Methods

To identify slow Methods

1. In the Time Frame list, select the period of time you want to analyze.
2. In the Instance list, select the instance you want to analyze.
3. Sort the tree according to the performance parameter you want to investigate.
4. Select the Method you want to analyze.
5. Continue expanding additional tree levels until you can identify the slow Methods.

Identifying the most invoked Service Request or Method

To identify the most invoked Service Request or Method

1. In the Time Frame list, select the period of time you want to analyze.

2. In the Instance list, select the instance you want to analyze.
3. On the Precise bar, go to **Settings>Tree View Settings**.
4. In the Tree View Settings dialog box, in the Sort by drop-down list, select **Executions**. Click **A** to display in Ascending order or **D** to display in Descending order.
5. Select the Method you want to analyze. Methods with the highest number of executions are most frequently invoked.

Searching for a specific Service Request or Method

To search for a specific Service Request or Method

1. In the Time Frame list, select the period of time you want to analyze.
2. In the Instance list, select the instance you want to analyze.
3. In the Tree pane, right-click the Service Request, Method, or instance node to search under.
4. From the popup menu, select the Search option.
5. In the Search dialog box, specify the search parameters by inserting the string to search for and the number of results to display.
6. Click **OK**. The results are highlighted in bold font and you can decide to focus on one of them by selecting it.