Oracle® JRockit Mission Control

Release Notes

3.1.0

April 2009

ORACLE®

Oracle JRockit Mission Control Release Notes, 3.1.0

Contents

General Release Information

Oracle JRockit Mission Control 3.1.0 Release Notes

Oracle JRockit Mission Control 2.0 and 2.0.1 Release Notes

JRockit Mission Control 1.0 Release Notes

General Release Information

This document contains important details for Oracle JRockit Mission Control. It contains information on the following subjects:

- Platform Support

- Installation

- Documentation Accompanying JRockit Mission Control

Platform Support

JRockit Mission Control is supported on the same platforms as Oracle JRockit JDK:

- Oracle JRockit JDK 6 (R27.2) is certified on the platforms listed on the following Supported Configurations page.

- Oracle JRockit JDK 5.0 (R26 and R27) are certified on the platforms listed on the following Supported Configurations page.

- Oracle JRockit JDK 1.4.2 (R26 and R27) are certified on the platforms listed on the following Supported Configurations page.

The JRockit Mission Control Client is currently only supported on Windows and Linux x86 (Linux with Mozilla 1.6 GTK2).

Installation

For instructions on installing Oracle JRockit Mission Control, please see *Installing Oracle JRockit Mission Control.*

Note: A separate license file is required for older versions of JRockit Mission Control but any technical license checks have been removed.

Documentation Accompanying JRockit Mission Control

The full documentation for JRockit Mission Control 2.0 and later is included as online help with the installation.

Documentation for JRockit Mission Control 1.0 is available online, see:

http://edocs.bea.com/jrockit/tools/index.html

Oracle JRockit Mission Control 3.1.0 Release Notes

Oracle JRockit Mission Control 3.1.0 is a further improvement of the JRockit Mission Control tools platform built on Eclipse Rich Client Platform (RCP) technology. These release notes contain important details about the latest enhancements and capabilities found in JRockit Mission Control 3.1.0. It contains information on the following subjects:

- JDK Update Versions Supported by this Release

- New Features and Enhancements in this Release

- New Features and Enhancements in Earlier Releases

- Most Recent Changes

- Known Issues

JDK Update Versions Supported by this Release

Oracle JRockit R27.6.3 supports these versions of the Jave JDK:

- Java 6 SE Update 11

- J2SE 5.0 Update 17

- J2SE 1.4.2_19

New Features and Enhancements in this Release

JRockit Mission Control 3.1.0 contains a large number of new features that will provide more information more seamlessly and improve the overall user experience. This section describes those features in the following subsections:

- General Product Enhancements
- Management Console
- JVM Browser
- JRockit Runtime Analyzer (JRA)
- Eclipse Integration
- New public APIs

General Product Enhancements

General enhancements to JRockit Mission Control provide:

- Accessibility Support
- Links to Mission Control Forum and Home Page
- Better Support for Copy and Paste

Accessibility Support

The Management Console, the JRockit Runtime Analyzer and the JVM Browser can be navigated by the keyboard and custom widgets in Mission Control, like graphs and pie charts, now provides information for screen readers. Graphs and dials in the Management Console can also be rendered as tables by changing accessibility options in Preference dialog box. For more information, please refer to Accessibility Notes for JRockit Mission Control Client at:

`http://download.oracle.com/docs/cd/E13150_01/jrockit_jvm/jrockit/tools/intro/a ccessibility.html`

Links to Mission Control Forum and Home Page

To help the user to find further assistance and information about Mission Control the help menu now contains links to the Mission Control Forum and Home Page .

Better Support for Copy and Paste

Copy and Paste now works for all tables and trees in the Management Console, the JRockit Runtime Analyzer and for the Properties View. Settings for turning column headers on/off and for indenting columns when copying trees have been added.

Figure 2-1 Better Support for Copy and Paste
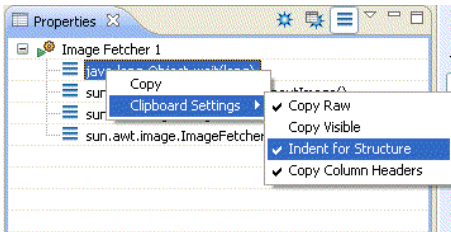
Management Console

New features in the JRockit Management Console include:

- Improved MBean Drilling

- Improved Dials

- MBean and Attribute Filtering

- Import/Export Trigger Rules

- Improved MBean Browser Tree Configuration

- GC Pause Time

- MBean Registration/De-Registration

- Thread CPU Usage

- Trigger Templates for WLS and JRockit

- Triggers Description

- MBean Metadata

- Toolbar navigation

- Diagnostic Commands Tab

● Garbage Collection Button

Improved MBean Drilling

The MBean Browser can now drill infinitely deep down into MBean attributes consisting of CompositeData, TabularData, Collections and arrays (Figure 2-2).

Figure 2-2 Improved MBean Drilling
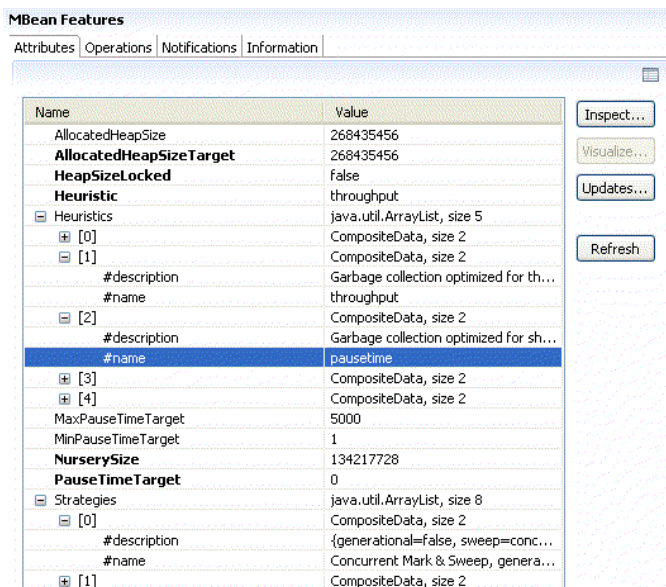
Improved Dials

The dials on the Overview Tab (Figure 2-3) show the current and maximum value. You can configure the background gradient and how values are formatted and add new dials to the dashboard.

Figure 2-3 Improved Dials
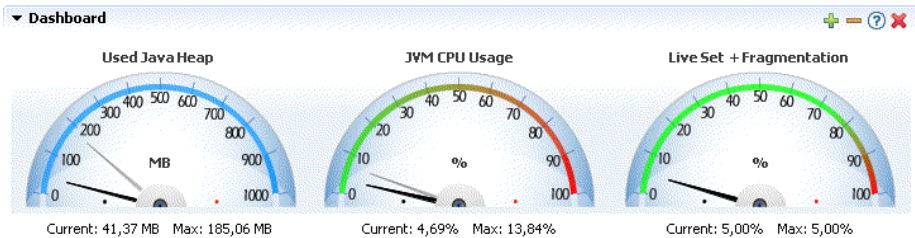
MBean and Attribute Filtering

The MBean Tree (Figure 2-4) and the Attribute Selector dialog box can be filtered using a wildcard expression.

Figure 2-4 MBean Filtering
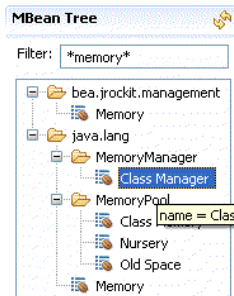
Import/Export Trigger Rules

You can now import and export trigger rules from and to the local file system, either by using the import/export wizard (Figure 2-5) in the file menu or by clicking on the import/export button on the Triggers Tab.

Figure 2-5 Import/Export Trigger Rules
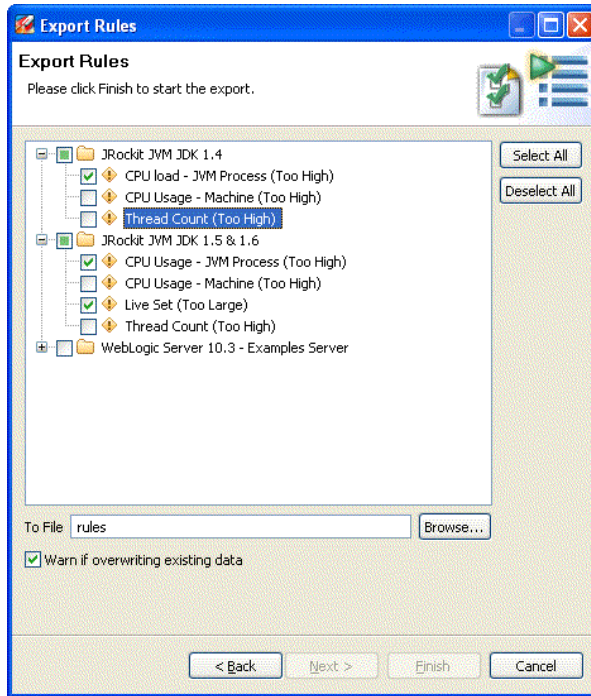
Improved MBean Browser Tree Configuration

Settings have been added for configuring how MBean names should be presented in the MBean Browser Tree (Figure 2-6).
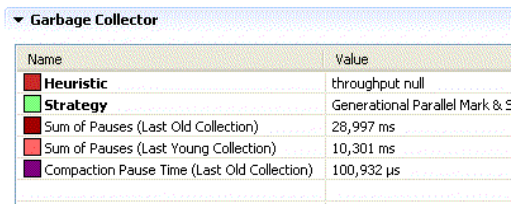
Figure 2-6 Improved MBean Browser Tree Configuration
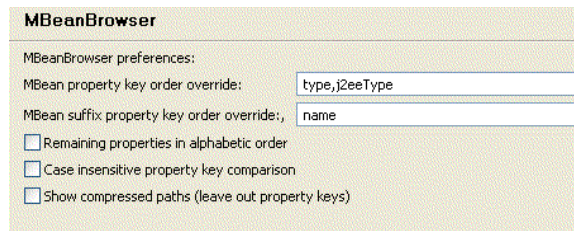
GC Pause Time

The total pause time for the last garbage collection can be viewed in the Memory Tab (Figure 2-7).

Figure 2-7 GC Pause Time

MBean Registration/De-Registration

The MBean Tree is refreshed when a new MBean is registered or unregistered on the monitored MBeanServer. You can turn on or off automatic refresh of the tree by using a toolbar button on top of the MBean Tree (Figure 2-8).
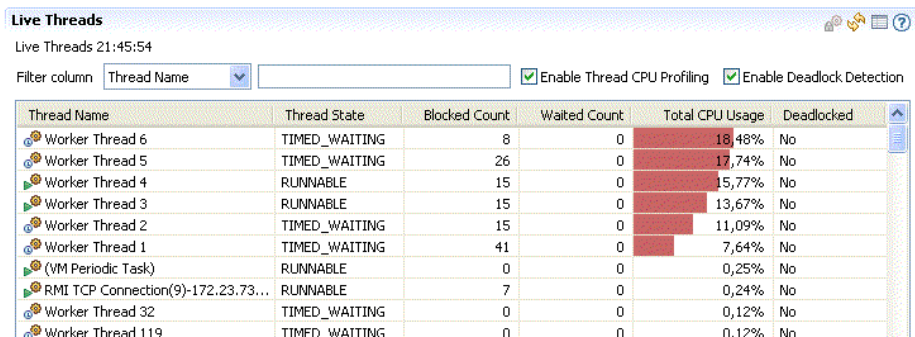
Figure 2-8 MBean Registration/De-Registration

Thread CPU Usage

The Threads Tab shows how much CPU a thread uses as a percentage of the total CPU usage on all cores (Figure 2-9).

Figure 2-9 Thread CPU Usage

Trigger Templates for WLS and JRockit

The Triggers Tab now contains a set of default rules that can serve as templates when creating triggers for JRockit or WebLogic 10.3 (Figure 2-10.

Figure 2-10 Trigger Templates for WLS and JRockit
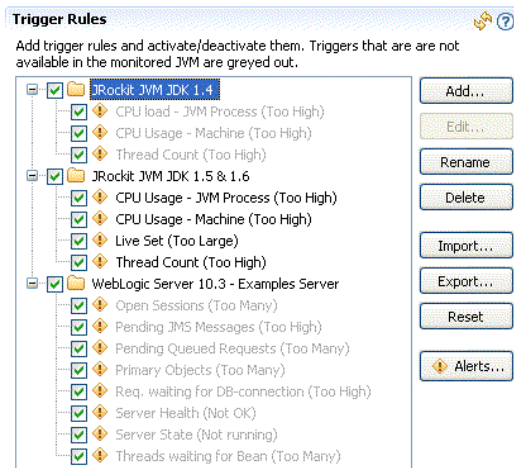
Triggers Description

When you create a new trigger, you can now add a description and format the text by using the `<B>`, `<LI>` and `<BR>`-tags (Figure 2-11).

Figure 2-11 Triggers Description

MBean Metadata

The MBean Browser Tab now shows MBean metadata (Figure 2-12).

Figure 2-12 MBean Metadata

Toolbar navigation

The tabs in the Management Console have been split into four groups: General, MBean, Runtime and Advanced and a toolbar (Figure 2-13) has been added so you can navigate between the

different groups. Tabs that have been added using the Mission Control tab extension point will be put in a fifth group called Other.

Figure 2-13 Toolbar Navigation
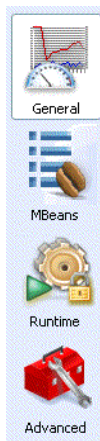
Diagnostic Commands Tab

The Management Console has a tab called Diagnostic Commands (Figure 2-14) that allows power users to execute Ctrl-break handlers from a remote machine over JMX.

Figure 2-14 Diagnostic Commands Tab
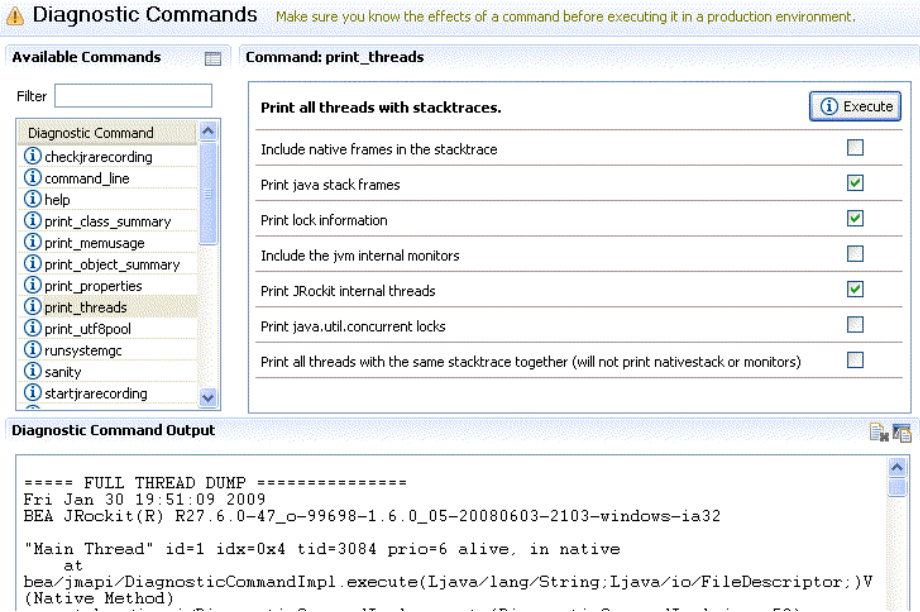
Garbage Collection Button

The Memory Tab contains a toolbar button for triggering a full garbage collection on the JVM that is being monitored (Figure 2-15).

Figure 2-15 Garbage Collection Button
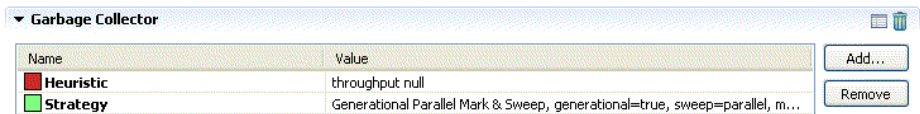
JVM Browser

New feaures in he JVM Browser include:

- Test Connection

- User Configurable Application Names

Test Connection

In the New Connection Wizard you can test a connection—to ensure that host name, port, password, and so on, are correct—before adding it to JVM Browser Tree (Figure 2-16).

Figure 2-16 Test Connection
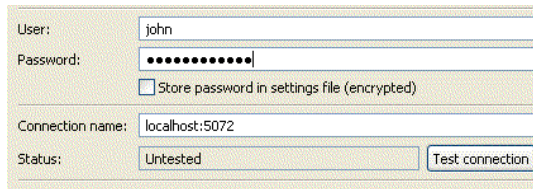
User Configurable Application Names

You can now, by using a set of naming rules, give discovered Java-applications human readable names in the JVM Browser (Figure 2-17).

Figure 2-17 User Configurable Application Names
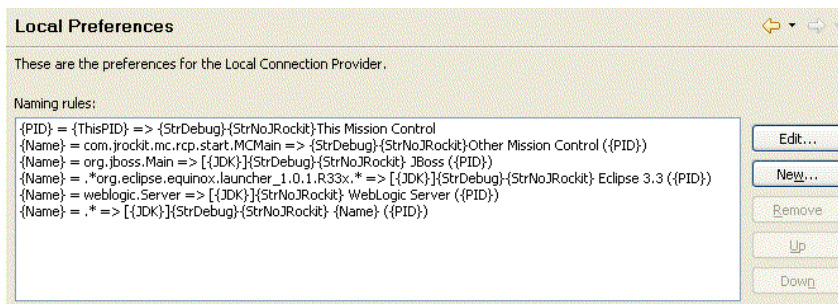
JRockit Runtime Analyzer (JRA)

New features in the JRA include:

- New Recording Templates
- Toolbar Navigation
- General Tab
- Code Overview Tab
- System Tab

- Recording Tab

- GC Statistics Tab

- Allocation Tab

- Threads/Locks Overview Tab

- Thread Tab

- Latency Overview Tab

- Latency Thread Tab

- Latency Histogram Tab

- Occupied Heap

- Improved Threads Tooltips

- Improved Tooltips for Pie Charts

- Configurable Tab Navigator

New Recording Templates

The JRA-recording wizard features four new templates:

- The *Full Recording* template is the default and it will try to gather as much information as possible without worrying too much about the overhead.

- The *Minimal Overhead Recording* template will try to minimize overhead even if it means that some data will not be recorded.

- The *Real Time Recording* template is the one to select when you use JRockit Real Time, as it will not provoke a garbage collection when gathering heap related information.

- The *Classic Recording* template is used for older versions of JRockit, which lack the support for recording latency related information or that use a JDK 1.4, which doesn't support file transfers back to Mission Control.

You can select the necessary template from the **Select recroding template** drop-down list (Figure 2-18).

Figure 2-18 New Recording Template Selection
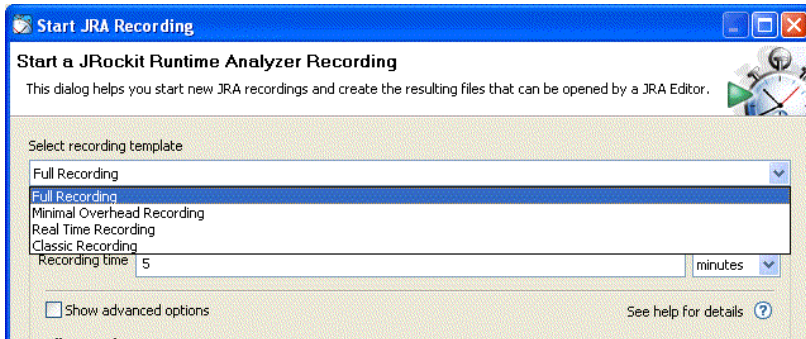
Toolbar Navigation

The tabs in the Runtime Analyzer have been split into six groups: General, Memory, Code, Threads/Locks, Latency and Other and a toolbar (Figure 2-19) has been added so you can navigate between the different groups.

Figure 2-19 JRA Toolbar
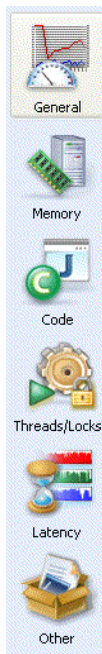
General Tab

There is a new overview tab that presents the most relevant data for the whole recording graphically (Figure 2-20).

Figure 2-20 General Overview Tab
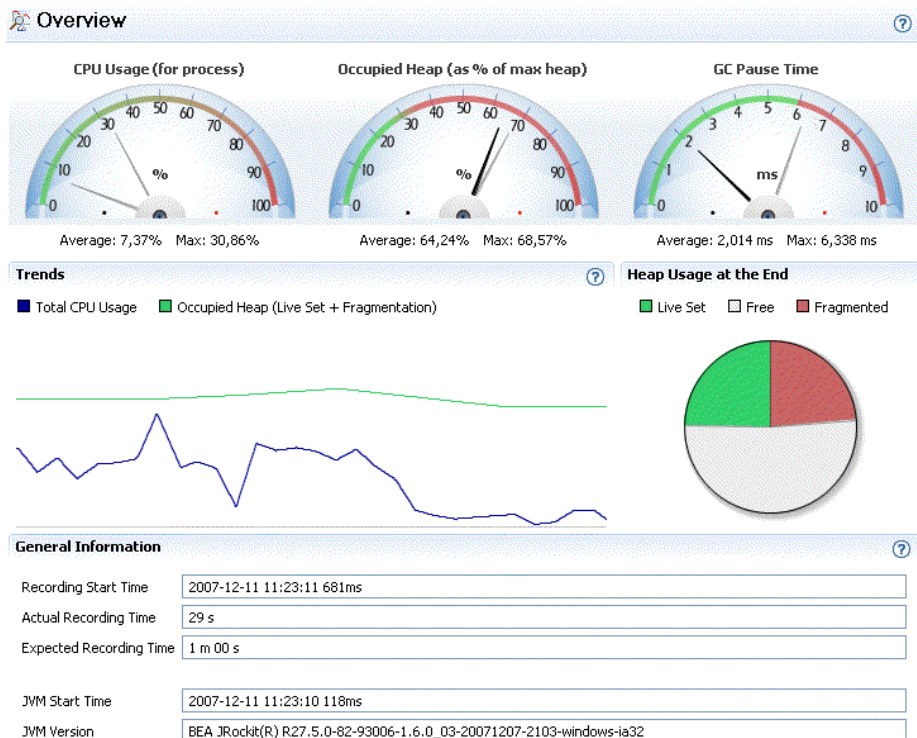
Code Overview Tab

There is a new overview tab for code related information that shows which packages and classes the application spent the most time executing (Figure 2-21).

Figure 2-21 Code Overviwe Tab
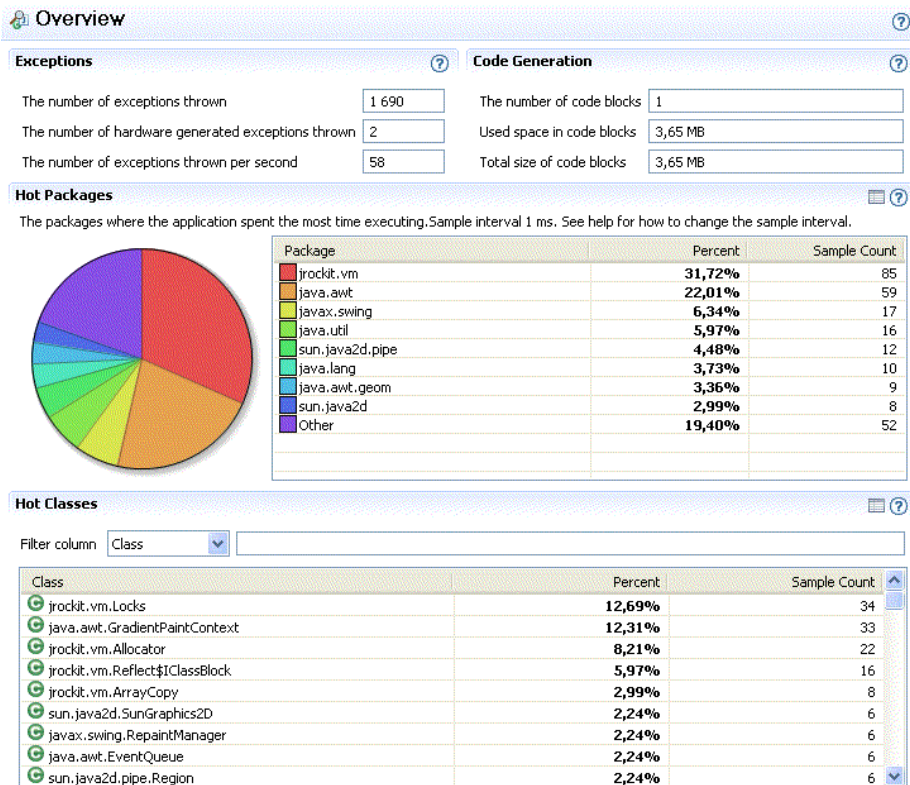
System Tab

There is a new System Tab that shows JVM and OS-related information (Figure 2-22).

Figure 2-22 System Tab
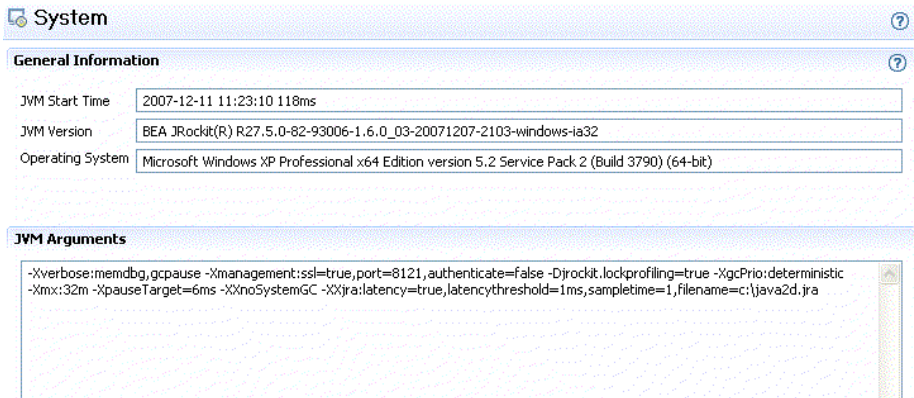
Recording Tab

There is a new Recording Tab that shows the start time of the recording and the recording parameters that were used(Figure 2-23).

Figure 2-23 Recording Tab

GC Statistics Tab

The GC General tab has been renamed to GC Statistics and it contains more detailed information about garbage collection pauses and garbage collections. (Figure 2-24).

Figure 2-24 GC Statistics Tab

Allocation Tab

There is a new tab that shows thread local area (TLA) information and the ratio between the number of bytes allocated by small and large objects in the application. It's also now possible to see how much memory each thread has allocated during the recording (Figure 2-25).

Figure 2-25 Allocation Tab
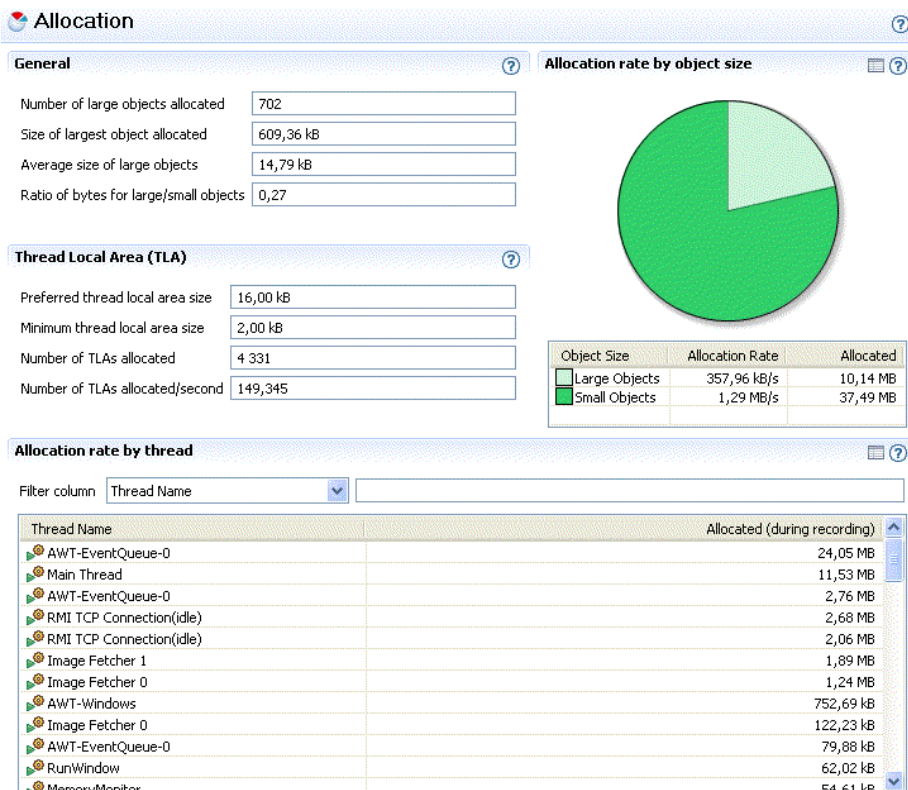
Threads/Locks Overview Tab

There is a new overview tab showing information about threads, locks and CPU usage(Figure 2-26).

Figure 2-26 hreads/Locks Overview Tab
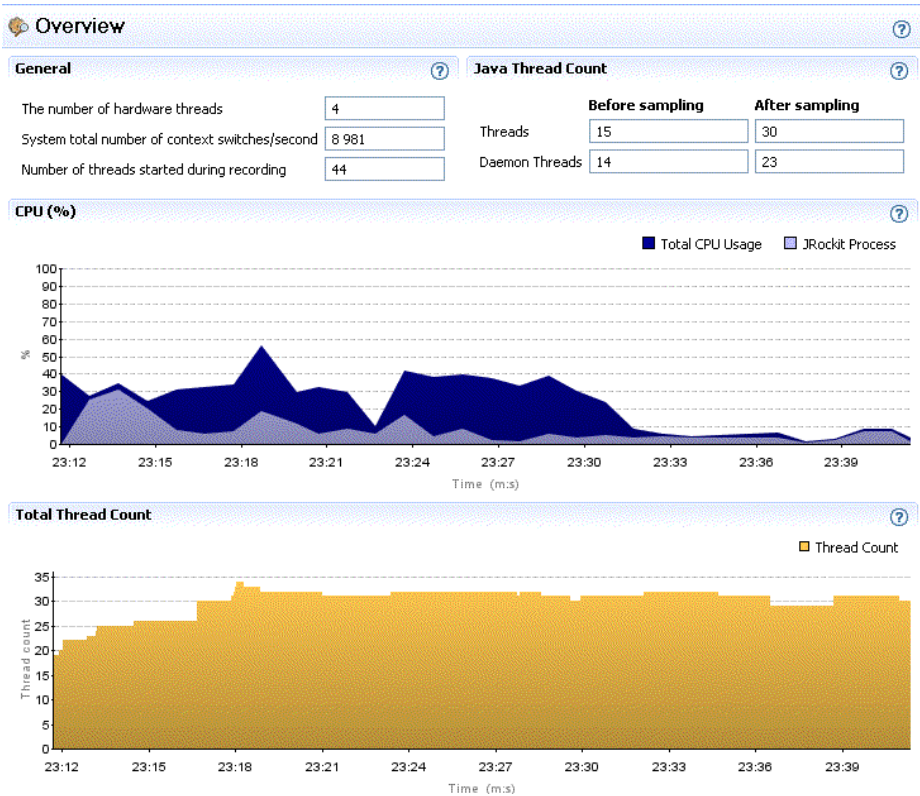
Thread Tab

There is a new tab that shows the threads that were running during the recording (Figure 2-27).

Figure 2-27 Thread Tab

Latency Overview Tab

There is a new overview tab displaying latencies categorized by type. The tab also shows which Java locks the application were blocked on the most (Figure 2-28).

Figure 2-28 Latency Overview Tab
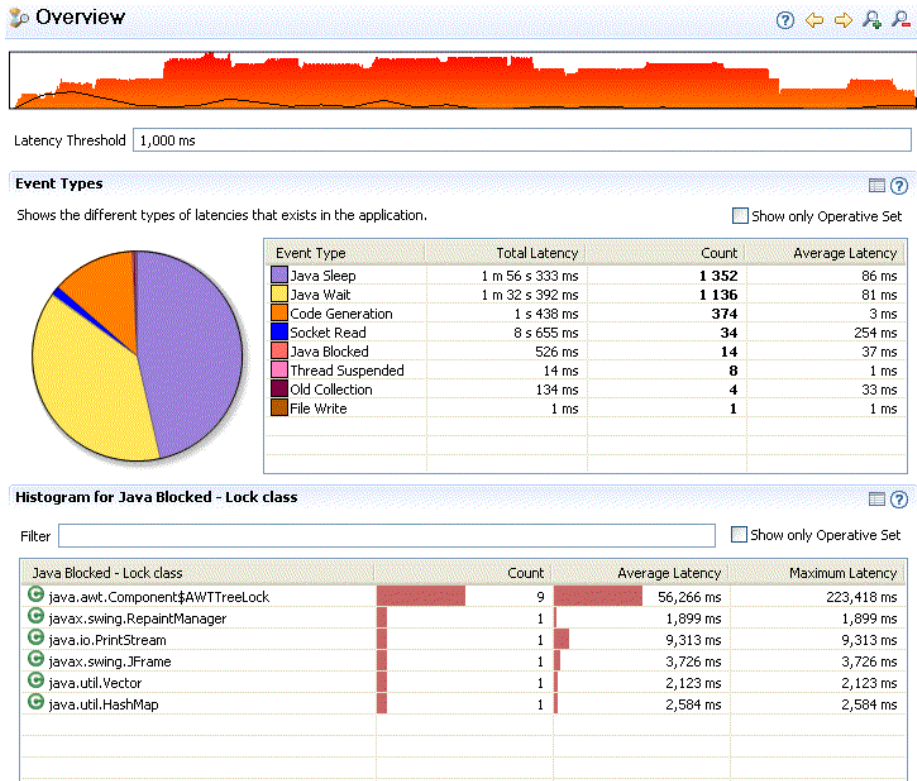
Latency Thread Tab

There is a new tab that shows the total and average latency per thread (Figure 2-29).

Figure 2-29 Latency Thread Tab
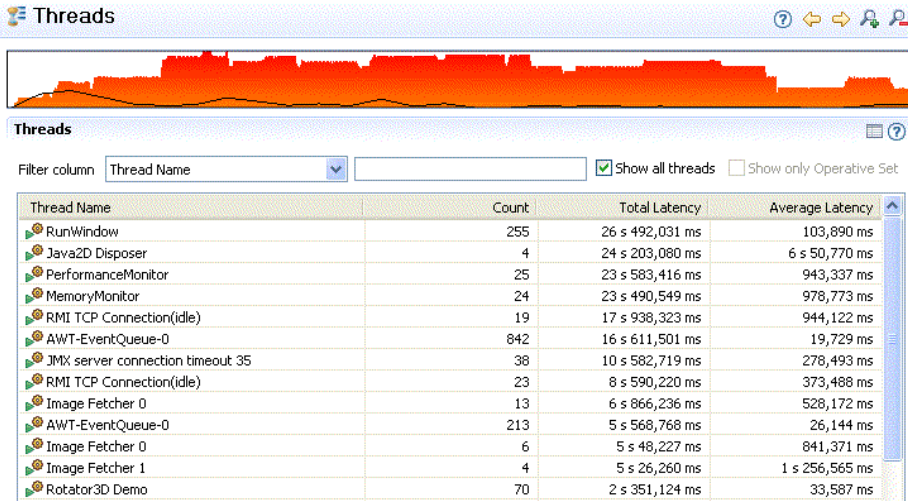
Latency Histogram Tab

The Event Value Histogram folder in the Latency Log Tab has moved to separate tab called the Histogram Tab and a details part, that shows the traces for all events that have a certain event value, has been added. This can be useful when trying to find out where in the application there has been contention on a lock (Figure 2-30).

Figure 2-30 Latency Histogram Tab
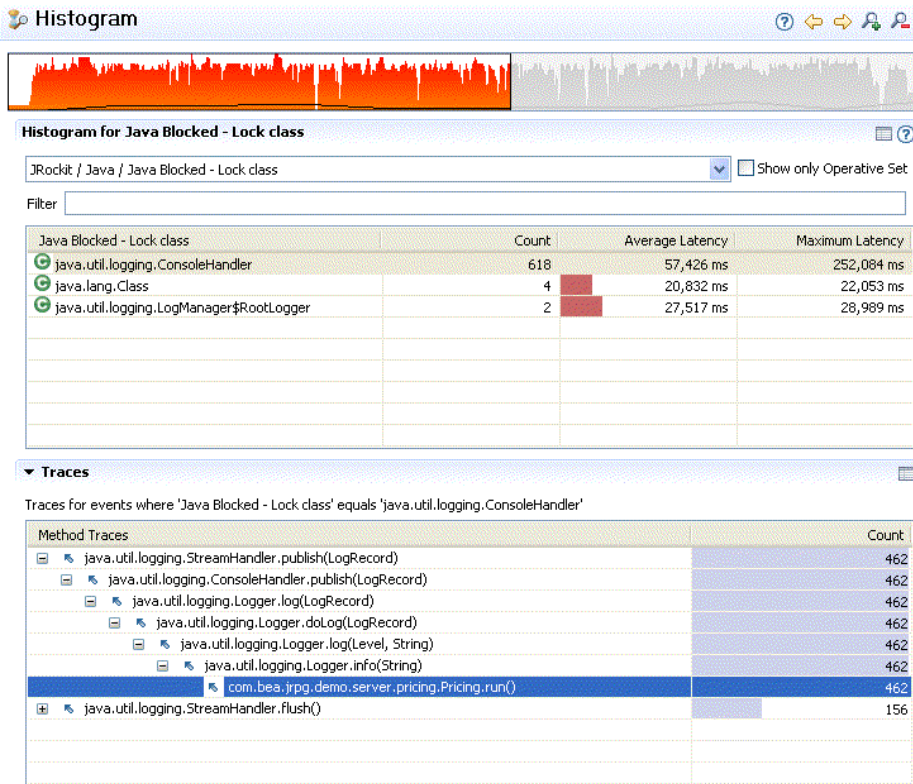
Occupied Heap

Occupied Heap, that shows the amount of memory after a garbage collection, has been added to GC Event Tab (Figure 2-31).

Figure 2-31 Occupied Heap
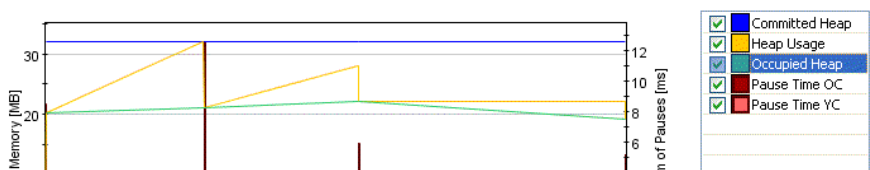
Improved Threads Tooltips

The tooltip for a thread in the Latency Graph now shows start and end time for the thread, the thread id and the total allocation rate (Figure 2-32).

Figure 2-32 Improved Threads Tooltips
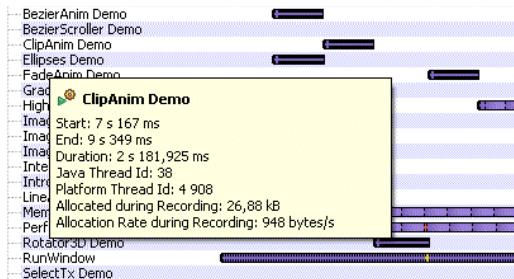
Improved Tooltips for Pie Charts

The tooltips for pie charts displays more detailed information about each slice (Figure 2-33).

Figure 2-33 Improved Tooltips for Pie Charts
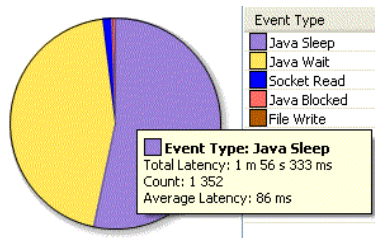
Configurable Tab Navigator

The position of tab navigator can now be configured by the user (Figure 2-34).

Figure 2-34 Configurable Tab Navigator
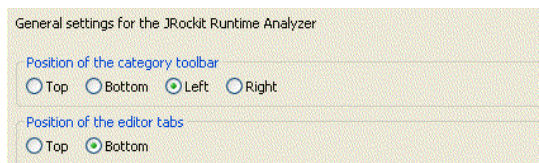
Eclipse Integration

New features in JRockit Mission Control's Eclipse version include:

- Tab Extension Point

- PDE Wizards

Tab Extension Point

There are now extension points that allow third parties to extends the Management Console with tabs, trigger rules and triggers actions (Figure 2-35.

Figure 2-35 Tab Extension Point
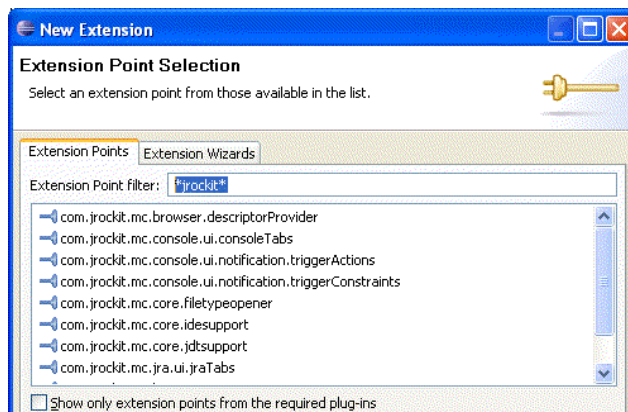
PDE Wizards

Eclipse wizards for creating classes that can use the new extension points exposed by JRockit Mission Control have been added (Figure 2-36).

Figure 2-36 PDE Wizards
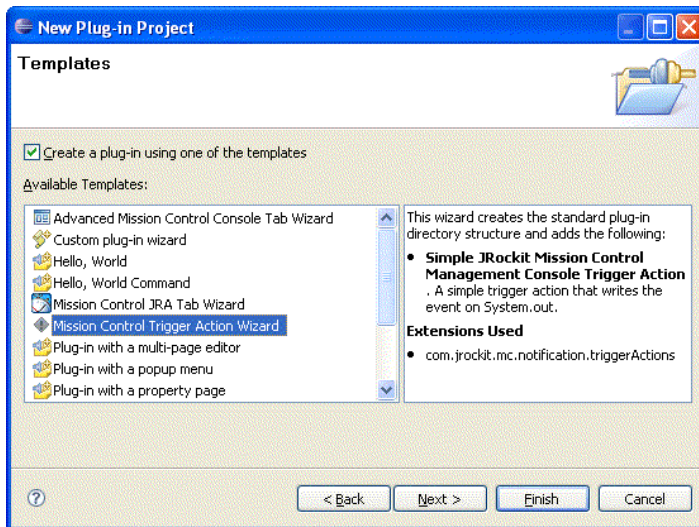
New public APIs

The RJMX-API, which is an extension of JMX that Mission Control uses to subscribe to JMX-data and to establish connections to local/remote servers, is now open for public use. The API also includes classes that allow communication with JRockit JDK 1.4, which relies on a JRockit specific protocol instead of JMX. Accompanying the RJMX-API are two Eclipse-plug-ins which exposes API-functionality that can be used to extend Mission Control and create dials, tables and charts similar to the ones found in Mission Control today.

New Features and Enhancements in Earlier Releases

Earlier versions of JRockit Mission Control include many new and useful features. These new features and changes to earlier versions of this product are described in these sections:

- About JRockit Mission Control 3.0.3

- Eclipse Integration of JRockit Mission Control 3.0.2

- Other JRockit Mission Control 3.0.2 Updates

- Latency Analyzer

- Recording Templates in JRA Recording Wizard

- Increased Recording Capabilities in the JRA

- User Interface is Localized

- Documentation is Localized

- Documentation is Available on eDocs as PDFs

About JRockit Mission Control 3.0.3

JRockit Mission Control 3.0.3 is a maintenance release and contains no new features. For a description of the changes in this release, please refer to Changes in JRockit Mission Control 3.0.3.

Eclipse Integration of JRockit Mission Control 3.0.2

The JRockit Mission Control Client is now available as an Eclipse plug-in edition. The plug-in version of the JRockit Mission Control Client provides seamless integration of JRockit Mission Control's application profiling and monitoring toolset with the Eclipse development platform. By integrating JRockit Mission Control with Eclipse, you will have easy access to the powerful toolset that comprises JRockit Mission Control.

When the JRockit Mission Control Client is run within the Eclipse IDE, you have access to IDE features that aren't otherwise available in the toolset when it is run as a standalone Rich Client Platform (RCP) application. The most significant of these features is the ability to see specific code in the running application by opening it directly from the JRockit Mission Control Client, a function called Jump-to-Source.

The other benefit of integrating the JRockit Mission Control Client with the Eclipse IDE is that it allows you to profile and monitor an application during its development phase just as you would during its production phase. This allows you to spot potential runtime problems before you actually deploy your application to production; for example, you might, while monitoring an application during its development notice a memory leak. By catching the memory leak during development, you can correct it before you migrate your application to a production environment.

For more information, please see Integration with the Eclipse IDE or open the JRockit Mission Control Client and launch the help system.

The location of the Eclipse update site will be published at `http://dev2dev.bea.com/jrockit/tools.html` when available.

Other JRockit Mission Control 3.0.2 Updates

- The JRockit Runtime Analyzer now shows the number of bytes of objects allocated by each Java thread.

- Three sample files that demonstrate the features of the Latency Analysis Tool have been added. The files are located at
  `JROCKIT_HOME/missioncontrol/samples/jrarecordings/`. The files are:

  - `pricing_server_logging_on.jra`

  - `pricing_server_logging_off.jra`

  - `java2d_demo.jra`

    Note: This file is a recording of the demo located at
    `JROCKIT_HOME/demo/jfc/Java2D`. The Java2D demo folder contains the source, allowing this recording to demonstrate Jump-to-Source (Jump-to-Source is only available when you are running JRockit Mission Control within Eclipse, as described in Eclipse Integration of JRockit Mission Control 3.0.2).

- Small adjacent Latency Analysis Tool (LAT) events of the same type are now clearly marked to make them easier to distinguish.

- Configurable velocimeters (Figure 2-37) have been added to the Console

Figure 2-37 Configurable Velocimeter
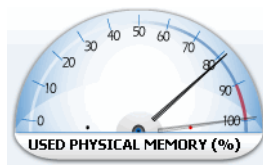
- You can see the exact numerical value for a point in a graph in a tooltip by hovering your mouse pointer at the point.

  Note: This feature is only available when the graph is frozen.

Figure 2-38 Displaying the Value for a Point on a Graph
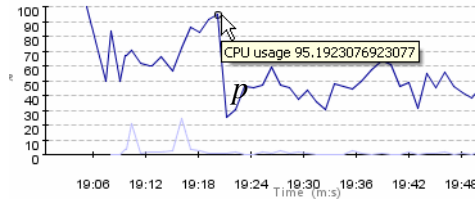
- The time ranges of graphs shown on the same page can be synchronized.

- You can filter attributes by name in the attribute browser when you select attributes to add to a graph or similar.

- Thread transitions—a latency event in one thread that is associated with another thread— are now displayed as small black arrows on the Latency Graph, as shown in Figure 2-39. By hovering your pointer over a transition arrow, a tooltip will appear, describing the transition.

Figure 2-39 Arrows Depict Thread Transitions in LAT; Tooltip Describes Selected Transition
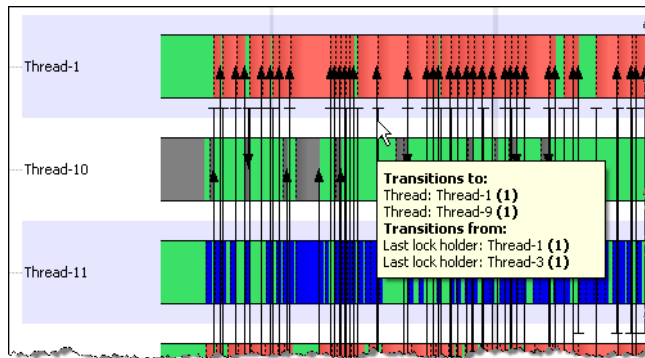
Latency Analyzer

A latency analyzer has been added to the JRA. You can create recordings that contain latency information for your application. The JRA Tool now contains three additional tabs that all show latency data from different perspectives. These tabs are prefixed *Latency* and named: *Latency Log, Latency Graph*, and *Latency Traces* (Figure 2-40). Together with these three tabs, there are two auxiliary tabs that allow you to turn on and off event types on the latency tabs and view properties.

All tabs prefixed with *Latency* share a common **Latency Timeline** slide bar where you can easily zoom in and out of your JRA recording to find latency events within a specific time frame.

Figure 2-40 New Latency tabs in JRA Tool
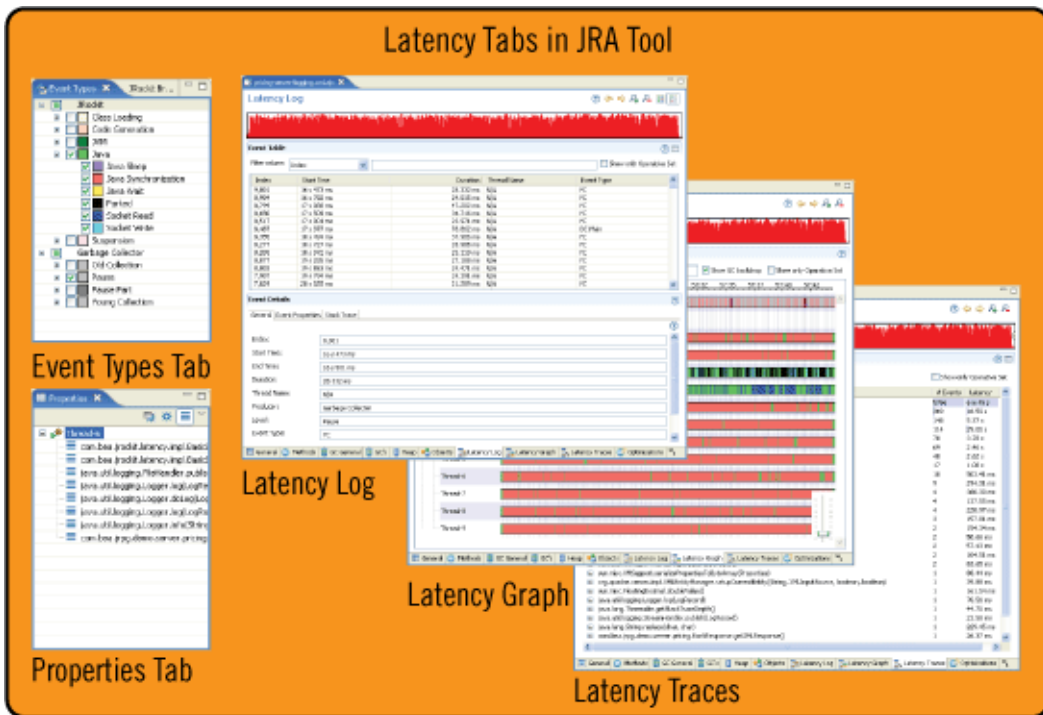
The new latency analyzer in the JRA Tool includes the following:

- **Latency Log** tab: the **Latency Log** tab lists the latency events that took place during the recording. By looking at latency data in the **Latency Log** tab, you can easily find a specific event type or select an attribute by using the sort and filter functions.

- **Latency Graph** tab: the **Latency Graph** gives you a graphical overview of how the application executes and it is easy to select events in terms of when they happened and in which thread.

- **Latency Traces** tab: the **Latency Traces** tab contains a list of all methods that contain latency events. The method traces with the most latencies are listed first. The **Latency Traces** table can be customized to display specific packages, classes, and methods.

- **Event Types** tab: the **Event Types** tab lists the events in relation to where they come from. The *Event types* themselves come from a *Level* and the Level comes from a *Producer*.

- **Properties** tab: the **Properties** tab lists the event properties, the event's stack trace, or the general event data depending on the view you have chosen.

Note: For older versions of JRockit Mission Control you will need a license file to use the Latency Analyzer. You can purchase the license from Oracle.

Recording Templates in JRA Recording Wizard

The JRA recording wizard now contains three templates that will make it easier to setup your JRA recording. The templates are the following:

- **JRA Recording Normal**—classic style to create a JRA recording.

- **Latency Recording Normal**—used to create latency data.

- **Latency Recording Minimal Overhead**—used to create latency data. It yields minimal overhead when recording latency data.

Figure 2-41 JRA recording wizard templates
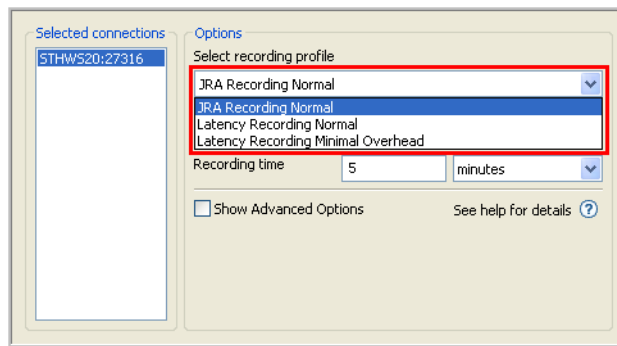
Increased Recording Capabilities in the JRA

In addition to the new Latency Analyzer recording capabilities, other recording capabilities have been added to the JRA: thread dumps and CPU load can be specified under the advanced options when creating a JRA recording. You can also set the interval for each sample type (Figure 2-42).

Figure 2-42 Thread dumps and CPU sampling in JRA recording
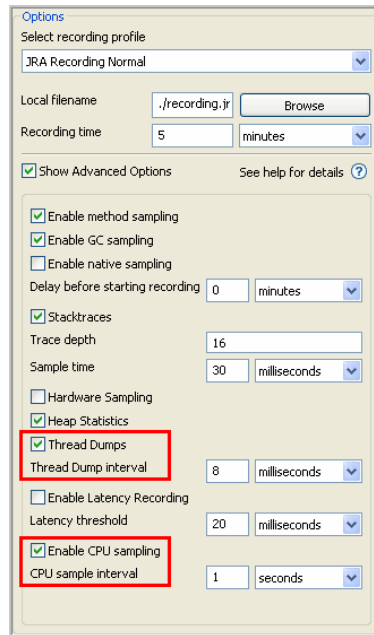
The JRA also records lazy unlocking statistics as part of lock profiling. For more information on lazy unlocking, see `-XXlazyUnlocking` in the Oracle JRockit JDK *Command-Line Reference*.

Possibility to View Thread Dump Data

In JRockit Mission Control 3.1.0 it is possible to record thread dump data in the JRA and then view thread dumps in the newly added Threads tab in the JRA Tool (Figure 2-43).

Figure 2-43 Thread dump in the JRA Tool
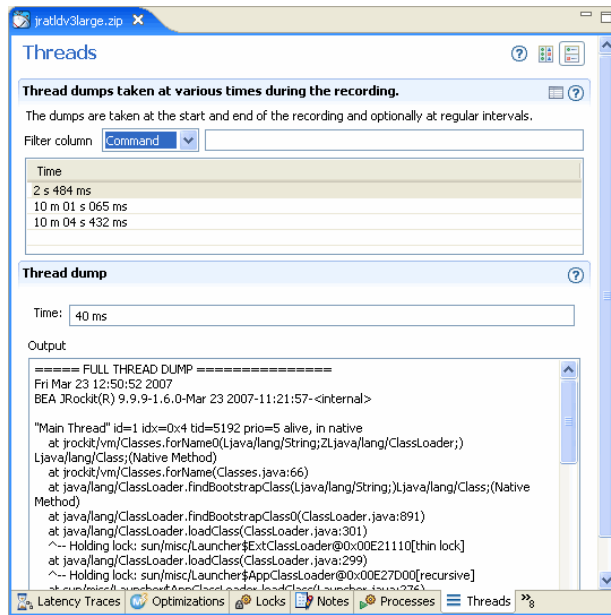
User Interface is Localized

JRockit Mission Control 3.1.0 is now available in a Japanese version (Figure 2-44) and a simplified Chinese version (Figure 2-45).

Figure 2-44 Japanese JRockit Mission Control
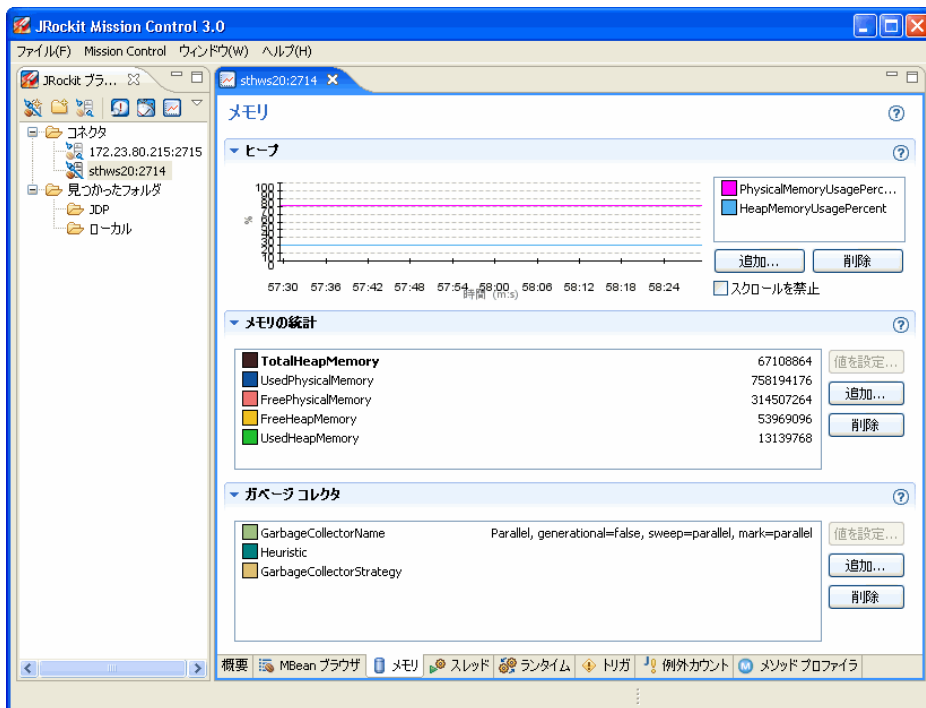
Figure 2-45 Simplified Chinese JRockit Mission Control
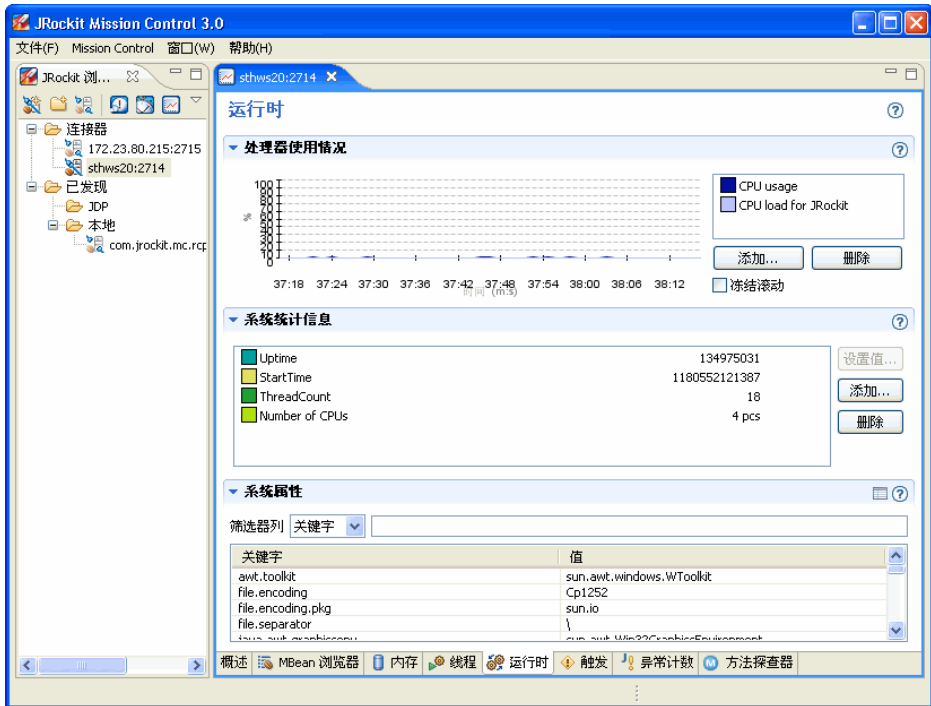
Documentation is Localized

The English user documentation for JRockit Mission Control 3.1.0 will be translated after the general availability release to Japanese and Simplified Chinese. Please refer to the Japanese and Chinese eDocs sites for more information.

Documentation is Available on eDocs as PDFs

For the JRockit Mission Control 3.1.0 release, the user documentation is available as online help within the tool itself and on eDocs as PDFs, see http://edocs.bea.com/jrockit/tools/index.html.

Most Recent Changes

This section describes the changes and issues resolved in all versions of JRockit Mission Control from 3.0.0 to 3.1.0.

- Changes in JRockit Mission Control 3.1.0

- Changes in JRockit Mission Control 3.0.3

- Changes in JRockit Mission Control 3.0.3

- Changes in JRockit Mission Control 3.0.1

- Changes in JRockit Mission Control 3.0.0

Changes in JRockit Mission Control 3.1.0

Table 2-1 lists changes in JRockit Mission Control 3.1.0

Table 2-1 Changes in JRockit Mission Control 3.1.0

Issue ID	Description
8294999	The error message that appears when you try to connect a 32-bit JRockit Mission Control instance to a 64-bit JVM has been updated to include "Unable to attach to 64-bit process".
8187650	Starting with Oracle JRockit Mission Control 3.1.0 and Oracle JRockit Real Time 3.1.0, each product installation directory is also an Oracle home, allowing additional product registration with the Oracle Central Inventory. The Central Inventory contains information about all installed Oracle products on the same host and can be managed by running the Oracle Universal Installer.
8179196	Starting with Oracle JRockit Mission Control 3.1.0 and Oracle JRockit Real Time 3.1.0 the demo and sample programs and the source code of the Java platform are no longer installed by default. They have been separated into optional components that explicitly must be selected by the user in order to be installed.

Changes in JRockit Mission Control 3.0.3

Table 2-2 lists changes in JRockit Mission Control 3.0.3.

Table 2-2 Changes in JRockit Mission Control 3.0.3

Issue ID	Description
CR369269	When viewing the GCs tab of a JRA recording, the legend for the References and finalizers chart has been changed. The label **Objects with finalizers** (formerly **Final references**) shows the total number of existing objects having a finalizer. The label **Finalization queue length** (formerly **Objects with finalizers**) shows the total number of objects waiting on finalization after a collection.
	In the Details - Old Collection section, the three corresponding labels (**# Objects with Finalizers**, **Finalization Queue Length Before**, and **Finalization Queue Length After**) have also been changed.
	The rationale for these changes is to better explain the values and what they represent.
CR367378	When trying to run the Eclipse version of JRockit Mission Control with Sun Hotspot JDK6_5, some users were unable to create a new connection and the connector panel showed only a few, if any, JDP connectors. This has been fixed.
	Note: Part of this fix is that, going forward, if you run JRockit Mission Control on any JVM other than JRockit JVM, a message box will appear, stating that running JRockit Mission Control on a non-JRockit JVM is unsupported.
CR364111	In earlier versions of JRockit Mission Control, the Uptime counter in System Statistics panel of the Management Console's Runtime tab was not working as designed. This has been fixed.
CR363616	When a trigger rule was active in multiple consoles, the internal state of the trigger in each console could be affected by data from other consoles. This has been fixed.
CR362074	A tooltip has been added to the Latency Time Line (the graph at the top in a LAT recording) that shows activity at a specific point on the timeline (Figure 2-46).

Figure 2-46 Latency Time Line Tooltip

Table 2-2 Changes in JRockit Mission Control 3.0.3

CR358674	The Latency Analysis Tool in JRockit Mission Control will now record threads that wait for object allocations or TLA fetches.
CR355590	When expanding references in the Memory Leak Detector's instance graph, almost every object node seemed to be referenced by a Global JNI Handle. Also, occasionally, objects would be referenced by the thread root (`Memleak Socket Reader`). This issue would clutter the graph with false dependencies, making it difficult to follow the actual reference chain and find the leak. This has been fixed.

Changes in JRockit Mission Control 3.0.2

Table 2-3 lists changes in this version of JRockit Mission Control.

Table 2-3 Changes in JRockit Mission Control 3.0.2

Issue ID	Description
CR359828	In the version of JRockit Mission Control that shipped with JRockit JDK R27.4, the value for **Heap Usage Before** for a garbage collection in a JRA recording was incorrect, as it actually showed the **Heap Usage After** for the proceeding collection instead. This is now fixed.
CR355927	When using the LAT with a JRA recording, the event start and end times were often incorrect; for example, while the correct time should be between `1min` to `1m30s`, user might have seen something like `13480days`.This has been fixed.
CR355308	Previously, if JRockit Mission Control was run with a Japanese or Traditional Chinese locale on an installation of Windows where the system font did not in itself contain glyphs for that language (such as in English editions of Windows, by default), bold fonts in the Memory Leak Detector would incorrectly be rendered as boxes. This has been fixed in R27.5.0.

Table 2-3 Changes in JRockit Mission Control 3.0.2

Issue ID	Description

CR353505	When you would specify a delay before the recording should start in the JRA recording wizard, the delay was not handled properly; for example:
	
	• The progress meter didn't take the delay into account when displaying the remaining time and percentage.
	
	• When you'd click the stop button before the recording has started
	   – You got an exception because the JRA editor could not download and save the file locally.
	   – The actual recording in JRockit JVM/JRAMBean was not aborted, so a new JRA recording could not be started.
	
	This has been fixed.

CR352607	When adding the **Sum of Pauses** from the table on the GCs tab, the total would differ from the total calculated from the same values in Excel. This has been fixed.
CR352424	The JRA now records the nursery size before and after young collections and shows these values in JRockit Mission Control.
CR351976	In releases prior to JRockit Mission Control 3.0.2 the columns **Generational**, **Mark Phase** and **Sweep Phase** in the GC General Tab in the JRA referred to the strategy that was used before the current strategy change. With JRockit Mission Control 3.0.2, these columns now refer to the strategy the garbage collection changed into. If recordings from a JRockit JDK release prior to R27.5 are opened up in JRockit Mission Control 3.0.2, some columns will show N/A. This is because all information about strategy changes were not recorded with these releases.
CR351646	JRA recording files will now always have the .jra file extension, regardless of what filename was used when starting the recording and whether or not compression was used.
CR345579	Previously, in some circumstances, JRockit Mission Control used more memory than necessary, causing the application to become unresponsive. This was most notable when opening several JRA recordings that had a lot of latency data, since JRockit Mission Control held on to data from recently opened recordings. The workaround was to restart JRockit Mission Control. This has now been fixed.
CR338771	On the Latency Log tab, if you copied one or more events from the Event Table, when you tried to paste the copied data, only the column headers would appear. This has been fixed.

Table 2-3 Changes in JRockit Mission Control 3.0.2

Issue ID	Description
CR330372	**New GUI Feature:** A command to export charts as images was added to the context menu for all charts displayed on the Management Console.
CR330371	**New GUI Feature:** A **Synchronize Charts** command has been added to the context menus of the charts on the Overview tab. This command sets the X-ranges of the other charts on the tab to the range X-range of the chart that triggered the context menu.
CR330371	**New GUI Feature:** A command to synchronizes the x-ranges of the other charts that triggered was added to the context menu on the Overview tab
CR330369	**New GUI Feature:** When you freeze a console chart and hoover the mouse pointer over the chart, the chart data value plotted closest to the pointer is circled and the value appears in a tool tip.
CR329869	Filtering has been added to the chart's attribute selectors to filter attributes based on their names and simplify attribute selection.
CR327148	The Growth column on the Trend tab in the Memory Leak Detector was showing a growth rate of zero bytes/second until at least two old space garbage collections had been performed. However, for some combinations of applications, heap size, and garbage collection algorithms, old space collections were performed so rarely that the growth rate never seemed to show anything other than zero. This has been fixed.
CR327148	The memory leak trend information wasn't updated properly when a generational concurrent garbage collector was used. This has been fixed.
CR319963	In earlier versions of JRockit Mission Control, if you started a JRA recording then pressed the stop button during the recording, JRMC would not show the data from the aborted file. Now, when recording with a R27.5 JRockit JVM or later (or with a R26.x JRockit JVM) if you stop a recording prematurely, it will be downloaded and opened in JRockit Mission Control.

Changes in JRockit Mission Control 3.0.1

Table 2-4 Changes in JRockit Mission Control 3.0.1lists changes in this version of JRockit Mission Control.

Table 2-4 Changes in JRockit Mission Control 3.0.1

Issue ID	Description
CR334175	The GC Duration attribute displays the complete garbage collection duration, not just the pause times. In this version of JRockit Mission Control, the former **Pause Time** field has been renamed **GC Duration**.
CR331770	In previous versions, the JRockit Mission Control MBean Browser did not handle quoted values in MBean ObjectNames well, especially not quoted values containing commas. When encountering such an MBean, the MBean browser would not be displayed. Such an MBean can, for example, be found in WebLogic Event Server. This problem is now fixed.
CR330569	In this version of JRockit Mission Control, the Persistence column from the MBean Browser's attribute browser has been removed and a command to persist graphs was added to the context menu.
CR326899	You can now filter latencies by type in the JRA Latency Trace tab. In R27.3 this kind of filtering was only available in the Latency Graph and the Latency Log tab.
	Stack trace filtering by latency type is a powerful technique that lets the user find out within seconds all the places in the code where the application had a latency problem of a certain kind, e.g. latencies caused by network I/O or latencies that happened because of lock synchronization.
CR326285	JRA recordings could sometimes contain illegal XML characters, which would cause an error when opening them in the GUI. The error typically looks like `org.xml.sax.SAXParseException: An invalid XML character`. This was caused by not properly escaping characters on the command line when writing the JRA file. This has been fixed.
CR325932	The GCs tab in JRA JRockit Mission Control now has a column called **Longest Pauses** which shows the longest pause that occurred during a garbage collection.
CR321958	You can now add event types/levels/producers to the operative set and change the color of event types and levels by using a context menu in the Event Types view.
CR321376	The Latency Analysis Tool has been augmented to cover read and write events from FileInputStrems, FileOutputStreams, FileChannels and RandomAccessFiles.

Table 2-4 Changes in JRockit Mission Control 3.0.1

Issue ID	Description
CR320149	This version of JRockit Mission Control has a command on an MBean Browser context menu that allows you to add selected attributes to an existing or new chart in the Overview tab.
CR302688	In earlier versions of JRockit Mission Control, you could not add more than one value to a chart at a time. In this version of JRockit Mission Control, the attribute selector associated with charts allows you to select and add multiple attributes to the chart in a single operation.

Changes in JRockit Mission Control 3.0.0

Table 2-5 lists changes in this version of JRockit Mission Control.

Table 2-5 Changes in JRockit Mission Control 3.0.0

Issue ID	Description
CR325754	The label *Pause Time* in the JRA Tool tab **GC's** has changed name to *Sum of Pauses*. This is because this number is the sum of several separate pauses within one garbage collection. To get more information about each garbage collection, look in the **Details** section on the **GC's** tab in the JRA Tool.
CR318629	Due to a bug in the attach framework (Sun bug #6559427), JRockit Mission Control was leaking several handles per locally-running JVM (JVM running on the same machine as JRockit Mission Control is) every time a JRockit Mission Control polls for locally running JVMs. This has been fixed in R27.4.
CR316739	The MBean Browser in the Management Console now uses property keys of ObjectNames to group MBeans. A preference setting has also been added that sets the MBean property key order.
CR315606	Now it is possible to properly disable native sampling, garbage collection sampling, and method sampling when connecting to a 1.4 version of JRockit JVM.
CR311136	JRockit Mission Control now displays folder names correctly when using Japanese characters.

Known Issues

Table 2-6 lists issues known to exist in JRockit Mission Control 3.0.0 through 3.1.0.

Table 2-6 Known Issues in JRockit Mission Control 3.0.0 Through 3.1.0

Issue	Description
8401283	Local JVMs might disappear from the JVM browser if the naming rules have been changed in the Local Preferences window (accessed from the **Window** menu by selecting **Preferences** > **JRockit Mission Control** > **Browser Preferences** > **Local Preferences**. If a rule is modified and does not follow the (undocumented) syntax in certain ways, and then applied, the thread that discover the local JVMs dies. **Workaround:** Correct all rules (by clicking **Restore Defaults**), apply them, and restart Mission Control.
8401232	In the dialog box used for entering a master password to encrypt JMX passwords, all text is missing; instead "NLS missing message:..." appears.
8400696	If you open a latency tab in a JRA-recording that doesn't contain any latency data, a help message is displayed. The documentation link in this help message doesn't work.
8323846	On occasion, a StackOverflow error can occur when you try to connect one JRockit Mission Control 3.1.0 to another instance of JRockit Mission Control 3.1.0.
8315572	Java applications running a 1.4 JVM will appear in the **Discovered/Local** folder in the JVM browser, prefixed with `[1.4]`. When trying to connect to one of these you will get an error message because local attach isn't supported in 1.4; however, this error message omits this important information: to connect to this JVM you should start the JVM with `-Xmanagement` and create a connection to `localhost:7090`. You also need to select the **JDK 1.4** radio button in the Connection Wizard's **Host JDK-version** group.
8311075	When reaching a tab component in the JRMC GUI, JAWS will instruct user "To switch pages, press Ctrl+Tab". The correct way to switch between tabs/pages is to use the left/right arrow keys.
8294981	If you try to start the Memory Leak Detector with a 32-bit JRockit Mission Control on a 64-bit JVM, you will receive an error message but it will not contain any information about JVM being 64-bit. Instead, the error message generated when you try to connect to the Console or start a JRA recording will contain this information.

Table 2-6 Known Issues in JRockit Mission Control 3.0.0 Through 3.1.0

Issue	Description
8294956	In some JRockit JVM versions, the `com.sun.management.HotSpotDiagnostic` MBean might appear in the Console MBean Browser even though it is not supported. Trying to view this MBean will display an `IllegalArgumentException` error message.
8289401	When starting a JRockit Mission Control JRA recording and selecting **Show advanced options**, if you've set a large font, low resolution, 1024x768 resolution with a two-line taskbar and so on, and the window is too large to fit the screen, you might not be able to see or reach all of the options.
8289326	JRockit Mission Control JRA contains some tables in the form of matrices of text boxes with row and column headers. When using a screen reader, these headers won't be read as they should be for a normal table.
8289304	When tabbing backwards in the properties dialog box for the JRockit Mission Control Console dashboard dials, you might get stuck in the Formatting combo box. Also, **Show Watermark** has no visible or audible focus.
8287549	To read JRockit Mission Control table data more efficiently with screen reading software, copy and paste the table data into a text editor and read it from there. To copy and paste, do the following: 1. Right-click the table you want to read to open the context menu. 2. Select all items by selecting **Select All**. 3. Select **Copy**. 4. Paste the text in a text editor.
8287358	To read data from the JRockit Mission Control Console, General/Overview tab with screen reading software, select **Window** > **Preferences** > **JRockit Mission Control** and choose **Render dials as tables** and **Render charts as tables** in the right panel.
8285833	If two or more threads have the same OS thread ID, only one of them will show up in the JRockit Mission Control JRA Latency graph. The likelihood of this occurring is quite low.
8178920	The Memory Leak Detector is not encrypted; therefore, you should never use it on a public network. An encryption protocol will be available in a future edition of this product.
CR371385	When using the XWIN X-Client to run the JRockit Mission Control client GUI remotely on a Linux system, you can move the scrollbars in JRockit Mission Control only by using the scrollwheel (thus only vertically).

Table 2-6 Known Issues in JRockit Mission Control 3.0.0 Through 3.1.0

Issue	Description
CR371205	Before starting JRockit Mission Control on Linux, you should have an X-Server running, otherwise the JRockit JVM might crash in GTK.
CR360265	If an application running on a JRockit JVM instance ends during an ongoing JRA recording, the JRA recording will not be transmitted to the JRockit Mission Control client; however the recording will be stored locally and its location will be printed on stderr. **Workaround:** Fetch the recording manually using the location printed on stderr.
CR359954	In the version of JRockit Mission Control that shipped with the JRockit JDK R27.4 the value shown in **Heap Usage Before** for a garbage collection in a JRA recording is incorrect. The value shown is actually the value of **Heap Usage After** for the proceeding collection. This will be fixed in the version of JRockit Mission Control that ships with JRockit JDK R27.5.
CR355941	When starting a JRA recording from JRockit Mission Control, you can cause incorrect time values if you edit the time before selecting the recording type. This usually happens when you switch back and forth between seconds and minutes. If you set the number of seconds to less than a minute you can lose your settings and the time value will become zero. **Workaround:** Be sure to set the recording type before editing the time values.
CR359029	When attempting to save a graph as an image on Linux systems (accessible from the graph context menu), the file dialog box might lack a filename input text field and the **OK** button might not respond. If this is the case, saving the graph as a .png file will not work, and pressing **Cancel** is your only option.
CR354035	When a trigger rule is active in multiple consoles, the internal state of the trigger in each console might be affected by data from the other consoles in which the trigger is active. The firing of the trigger might become unreliable.
CR356053	Due to a counter overflowing, the exception counting APIs in JMAPI and in the JRockit Management Console return negative values if too many exceptions (more than 2^{31}) have been thrown in the lifetime of the JVM.

Table 2-6 Known Issues in JRockit Mission Control 3.0.0 Through 3.1.0

Issue	Description
CR355912	When JRockit Mission Control shipped with the JRockit JDK R27.1 is run with a Japanese or Traditional Chinese locale on an installation of Windows where the system font does not in itself contain glyphs for that language (such as in English editions of Windows, by default), bold fonts in the Memory Leak Detector will be incorrectly rendered as boxes. This will be fixed in the JRockit Mission Control shipped with the JRockit JDK R27.5.
CR355590	In the instance graph of the Memory Leak Detector, false references from a Global JNI Handle might be shown. Any such reference where the tooltip says "Number of handles:1" might be false. Similarly, but less frequently, references from a Threadroot:(MemLeak Socket Reader) are always false. These references are temporary references used in the implementation of the Memory Leak Detector in JRockit Mission Control. They do not keep objects alive.
CR355306	If the default Windows temporary directory (`java.io.temp`) is on a FAT file system, some tools will not be able to discover local processes. These tools include JRockit Mission Control, jrcmd and jconsole. For security reasons, local monitoring and management is only supported if your default Windows temporary directory is on a file system that supports setting permissions on files and directories (for example, on an NTFS file system). It is not supported on a FAT file system that provides insufficient access controls.
CR351557	In some rare cases, you might get a Script Debugger error when launching the Online Help in JRockit Mission Control on Windows. This can occur if you have deselected the option **Browsing>Disable script debugging (Other)** in Internet Explorer and have a script debugger installed. If you click **No** when prompted, everything will work as designed. **Workaround** Open the Tools menu in IE and select **Internet Options...** Under the **Browsing** section, select **Disable script debugging (Other)**.
CR339469	Copying event information from the Thread Latency Log table to the clipboard does not work properly. Only the header information will be copied. This issue will be fixed in the JRockit Mission Control version that ships with JRockit JDK R27.5.0.
CR338731	Some events in the JRA latency recordings have their thread ID's set to 0. In particular, this applies to `JVM Event Wait->Signalling thread`, `Java Synchronization->Last holder thread` and `Java Synchronization->Holder thread`.

Table 2-6 Known Issues in JRockit Mission Control 3.0.0 Through 3.1.0

Issue	Description
CR337475	In a JRA recording, the number of allocated TLA (Thread Local Areas) is recorded, as well as the preferred size of a TLA (in bytes). The JRA GUI will multiply these values to get the number of bytes allocated in TLAs during the entire recording; however, the size of the TLAs actually used can sometimes be a bit smaller than the reported size (the preferred size is only a preferred size; fragmentation can cause the TLAs to become smaller) and the value printed in the GUI can be overestimated.
CR333156	Currently, if you are running JRockit Mission Control on a 1.4 version of the JRockit JVM shipped with the JRockit JDK R27.1, you cannot use the JRA Recording Wizard to start JRA Recordings with latency data. RMP (the legacy management protocol) does not currently provide information about the recording capabilities of a JRockit JVM instance. **Workaround** To start a JRA recording with latency data, please use on of the following workarounds: • Use the JRockit JDK tool jrcmd, as described in Creating Fully Featured JRA Recordings with 1.4 Based JRockit JVMs. • Start JRockit JVM with the command line option -XXjra.with the latency parameter set to true (this method is also described in Creating Fully Featured JRA Recordings with 1.4 Based JRockit JVMs).
CR326908	The label *Pause Time* in the JRA Tool tab **GC's** has changed name to *Sum of Pauses*; however, this change has not made it to the translated Japanese and simplified Chinese versions.

Table 2-6 Known Issues in JRockit Mission Control 3.0.0 Through 3.1.0

Issue	Description
CR323065	If JRockit Mission Control is started on RHEL4 using the `bin/jrmc` executable, the online help may not work. Instead of opening the help browser, a dialog saying "Couldn't open help on {0}" is shown.
	Workaround:
	1. Install a supported version of XULRunner, Firefox, or Mozilla.
	Instructions can be found at http://www.eclipse.org/swt/faq.php#browserlinux.
	2. Set the environment variable `MOZILLA_FIVE_HOME` to your `XULRunner/Firefox/Mozilla` installation folder, e.g. export `MOZILLA_FIVE_HOME=/usr/lib/mozilla`
	3. Set the environment variable `LD_LIBRARY_PATH` to include `MOZILLA_FIVE_HOME`, e.g. export `LD_LIBRARY_PATH=${MOZILLA_FIVE_HOME}:${LD_LIBRARY_PATH}`
	4. Run `bin/jrmc`.
	This problem is also described at http://www.eclipse.org/swt/faq.php#browserlinuxrcp
CR313460	When looking at predecessors in the **Methods** tab of the JRA Tool, sometimes the percentage can become lower, even though the nodes are not branching. This is caused by stack depth not being high enough for some of the samples participating in calls leading through the method being looked at. You can avoid this by increasing the Trace Depth in the JRA Recording Wizard.

Oracle JRockit Mission Control 2.0 and 2.0.1 Release Notes

Oracle JRockit Mission Control 2.0.1 is the new JRockit JVM tools platform built on Eclipse Rich Client Platform (RCP) technology. These release notes contain important details about the latest enhancements and capabilities found in this and previous releases. It contains information on the following subjects:

- Features and Enhancements in JRockit Mission Control 2.0.1

- Features and Enhancements in JRockit Mission Control 2.0

- Most Recent Changes

- Known Issues

Features and Enhancements in JRockit Mission Control 2.0.1

JRockit Mission Control 2.0.1 is a maintenance release, which means that no major features have been added. The most noticeable improvements are the following:

- Start Time and Uptime Information in JRA

- List of Running Processes

- Time Information About Optimization

- Improved Visualization of Garbage Collections

Start Time and Uptime Information in JRA

In the General Information panel on the **General** tab of the JRA, you can now see when the JRockit JVM process started and how long thereafter the JRA recording started (in the **Uptime before recording** field).

Figure 3-1 shows the new fields **JRockit start time** and **Uptime before recording** field on the **General** tab in the JRA Tool.

Figure 3-1 The JRockit JVM start time and uptime before recording
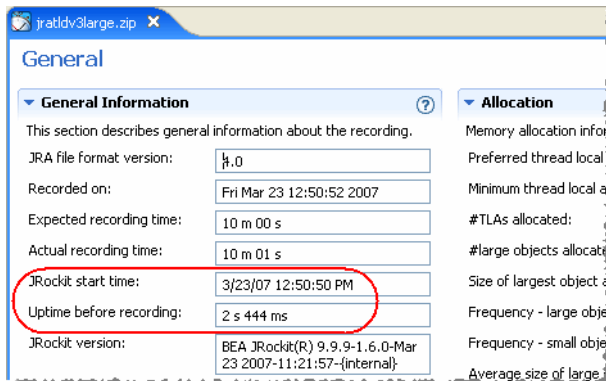
List of Running Processes

There is a new **Processes** tab in the JRA Tool (Figure 3-2). This tab gives you a snapshot of all processes running on the machine at the start and end of the recording (marked 1 in Figure 3-2). If you click on a process, you will see all its details (marked 2 in Figure 3-2).

Note: The command line information is not available for recordings made on Windows.

Figure 3-2 The new Processes tab in JRA Tool
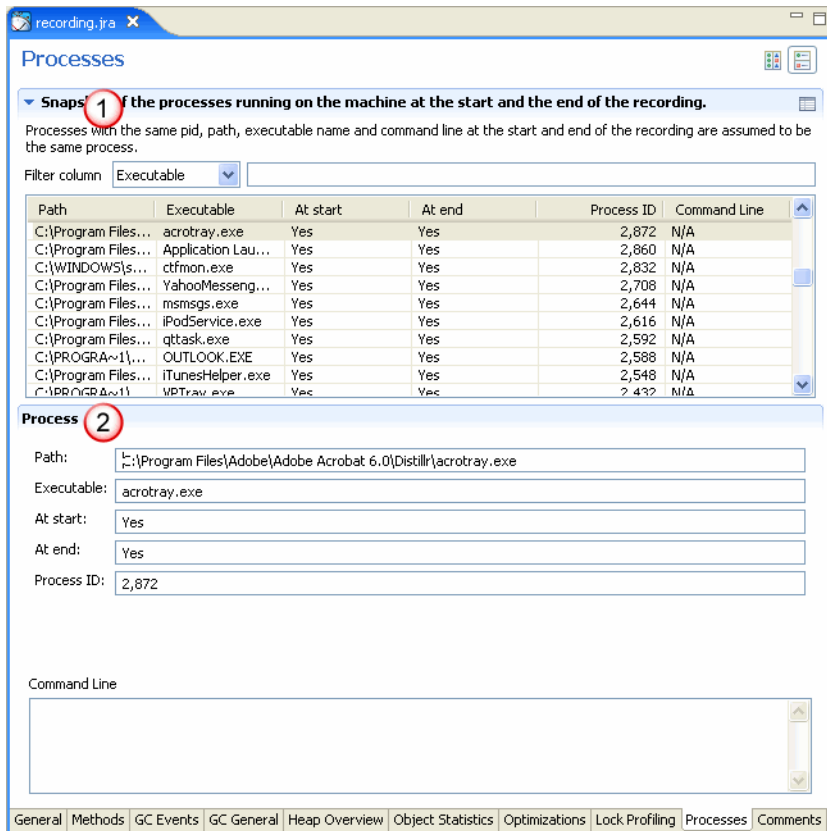
Time Information About Optimization

On the **Optimizations** tab in the JRA Tool (Figure 3-3), you can now see when an optimization of a method started (in the **Start Time** column), when it ended (in the **End Time** column), and how long it lasted (in the **Time** column).

Figure 3-3 The Optimizations tab in JRA Tool
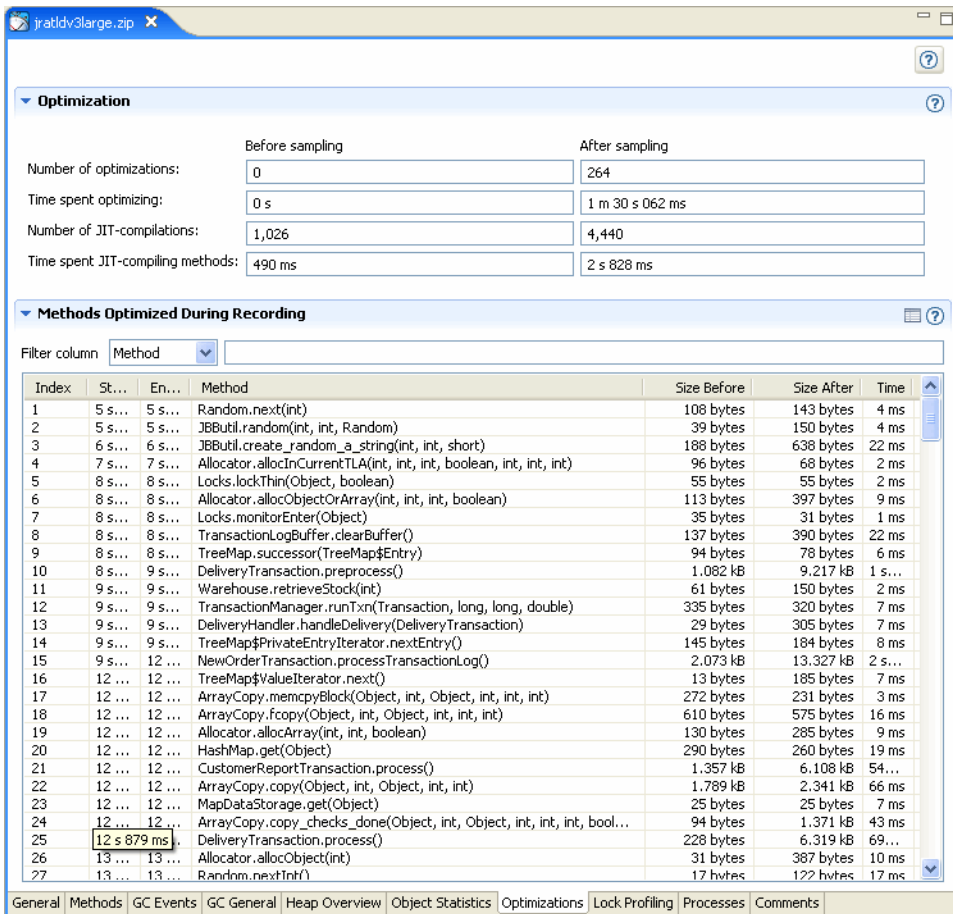
Improved Visualization of Garbage Collections

The pause times generated by old-generation garbage collections (Old Collections—OC) and nursery garbage collections (Young Collections—YC) are represented more clearly in the JRA Tool (Figure 3-4).

Figure 3-4 Old Collections and Young Collections in JRA Tool
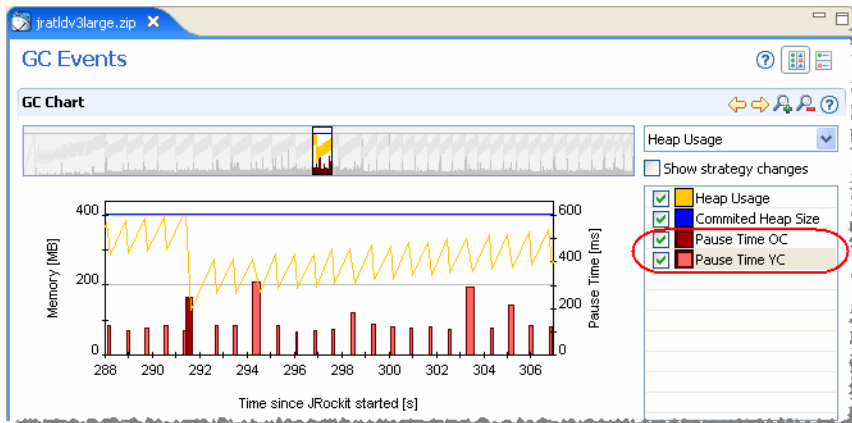
Features and Enhancements in JRockit Mission Control 2.0

For the release of JRockit Mission Control 2.0, there have been several major improvements to the tools. The most noticeable improvements are the following:

- New and Improved User Interface

- Usability Enhancements

- Additional Diagnostics Data

- Connect on Demand

- Changes in the JRockit Mission Control 2.0 Release

New and Improved User Interface

The user interface is now built on the Eclipse Rich Client Platform, which means that menus, mouse and keyboard behavior, the help system, etc. will be familiar to you if you are used to the Eclipse IDE.

All the familiar tools from the first version of JRockit Mission Control, have been merged into one unified GUI (see Figure 3-5) with the addition of a JRockit Browser that simplifies finding and connecting to running Java applications. It even allows you to monitor several applications

simultaneously. Figure 3-5 shows the new and integrated version of all available tools in JRockit Mission Control 2.0.

Figure 3-5 The JRockit Mission Control with all Applications Open
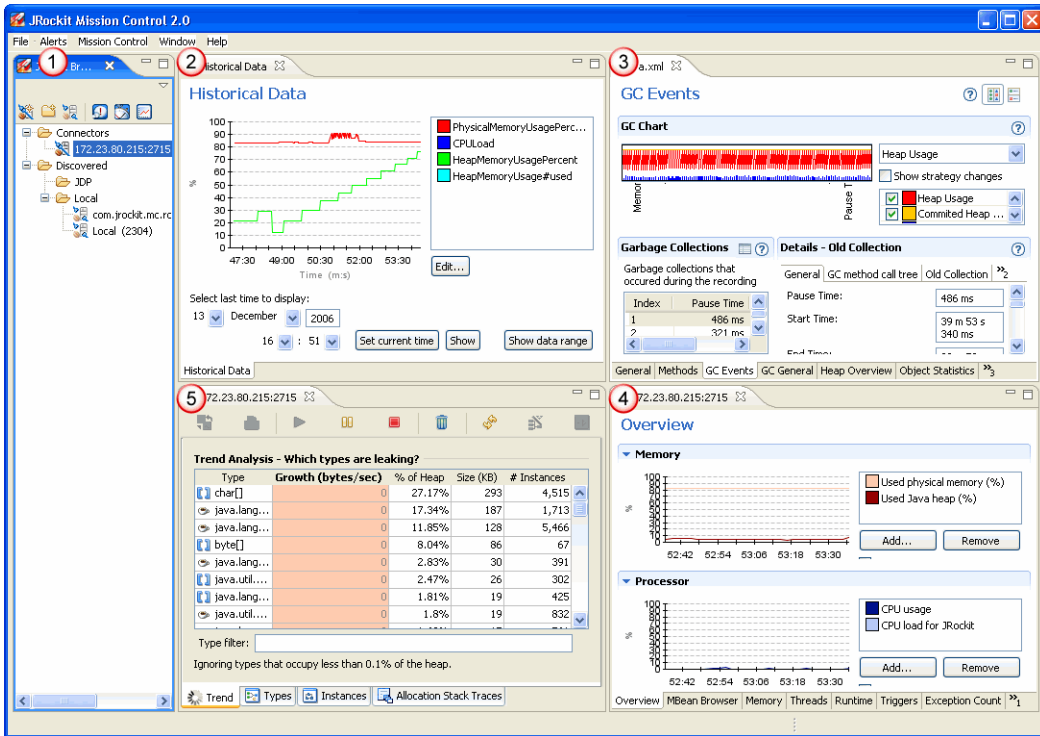
JRockit Mission Control 2.0 consists of the following parts:

1. **JRockit Browser**: the latest addition to the tools suite. A new way to locate a JRockit JVM instance and by selecting it start monitoring through the Management Console or the Memory Leak Detector.

2. **Historical Data**: a low overhead way to keep track of attributes and to see trends for attributes. You can select attributes from JRockit JVM, your own applications, or any other application that has MBeans registered on the MBean Server.

3. **JRockit Runtime Analyzer**: the popular runtime analyzer tool with several new additions: more data is recorded, filtering functionality in graphs, context sensitive help, and more.

4. **Memory Leak Detector**: the great way to spot memory leaks within your application. This tool is now included as part of JRockit Mission Control 2.0.

5. **Management Console:** the most widely used JRockit Mission Control tool that allows you to monitor live data from the JRockit JVM. This tool allows you to monitor the running Java application, create triggers that notify you when a predefined condition is met and much more.

Usability Enhancements

This release contains a large set of usability enhancements, for example:

- Drag and drop functionality.

- Documentation is now available from the **Help** menu. For the JRA tool, the help is context sensitive, which helps you analyze the immense amount of data available.

- Possibility to manipulate views freely, which for instance allows you to compare profiling records next to each other.

- Interfaces have been localized in Japanese. Use `jrmc -nl ja_JP` to set JRockit Mission Control to Japanese.

Additional Diagnostics Data

More detailed diagnostics data is now available in the JRA tool, in particular, more detailed GC information.

Connect on Demand

JRockit Mission Control 2.0, will automatically detect and be able to connect to the JRockit JVM (5.0 and later) instances on the local machine, as long as the JRockit JVM and JRockit Mission Control has been started by the same user.

For remote monitoring, the JRockit JVM includes the multicast-based JRockit Discovery Protocol (JDP), which allows JRockit Mission Control to automatically discover running instances of the JRockit JVM. When this is enabled, a automatically generated list of all discovered JRockit JVMs will appear in the JRockit Browser.

To enable JDP, start the JRockit JVM with `-Xmanagement:autodiscovery=true`, see -Xmanagement.

Most Recent Changes

The following has been fixed in the JRockit Mission Control 2.0 for the different release:

- Changes in the JRockit Mission Control 2.0.1 Release

- Changes in the JRockit Mission Control 2.0 Release

Changes in the JRockit Mission Control 2.0.1 Release

The following has changed in the JRockit Mission Control 2.0 release.

Change Request ID	Description
CR311881	The total number of context switches per second was calculated incorrectly on Linux. This has now been fixed. The number of context switches per second is presented in the **System total #context switches/second** field in the **Threads** panel on the General tab of the JRA Tool.
CR309660	In Oracle JRockit JDK R27.1 changing CPU affinity did not work for 1.4 versions of the JRockit JVM. It now works for all JDK levels.
CR307277	In JRockit Management Console, the **Freeze scrolling** check box is automatically selected when you zoom in on a graph by right-clicking and selecting **Zoom in** on the **Zoom** menu. Previously, this only happened when you zoomed in on a selection of the graph (by right-clicking and selecting **Selection** on the **Zoom** menu).
CR307277	Passwords were not stored properly from the JRockit Connection window in the JRockit Browser (when the **Store password in settings file (clear text)** check box was selected). This issue has now been fixed.
CR305851	The suggested port in the Connection Wizard for the Oracle JRockit JVM 1.4.2 is now 7090, that is, the default port for the RMP protocol used by the Oracle JRockit JVM 1.4.2. In JRockit Mission Control 2.0, the port had to be set manually to 7090 since the Connection Wizard suggested 7091 (the default port for the JMX protocol).
CR304563	Only file paths valid on the machine running JRockit Mission Control were allowed for JRA recordings (see Known Issues, CR304566). This has now been fixed.

Change Request ID	Description
CR303923	JMXMAPI did not properly support changing the garbage collection strategy by changing the strategy attribute of the `bea.jrockit.management.GarbageCollectorMBean` This has now been fixed. **Note:** The strategy changes when the next garbage collection occurs. You can force a garbage collection with the `gc` operation of the `GarbageCollectorMBean` after changing the strategy attribute.
CR303920	In the JRockit JVM included in JDK 5.0, attempting to start the management server on a port that was in use resulted in an exit from the JVM (see Known Issues, CR303920). This has now been fixed.
CR302945	The Compaction Ratio displayed in the Old Collections section on the GC Events tab in JRockit Runtime Analyzer (JRA) is now calculated correctly. The actual number of available compaction parts is always used in the calculation. Previously, the default number of available compaction parts was instead used in the calculation, sometimes resulting in an incorrect ratio.
CR302733	In the Management Console, you can edit the titles of graphs and axes. To do this, right-click on a graph and select the `Edit titles` submenu. On the submenu, select whether to edit the graph title, the X-axis title or the Y-axis title. When you have changed a title, the new title is stored and will be used when you open the Management Console again.
CR302635	It should not be possible to create, rename or delete folder nodes under the **Discovery** folder but these operations were enabled in JRockit Mission Control 2.0 (see Known Issues, CR302635). This has been fixed in JRockit Mission Control 2.0.1.
CR302571	Previously, in JRockit Mission Control 2.0, there were some limitations on the handling of JRA files (see Known Issues, CR302571) that have been fixed in JRockit Mission Control 2.0.1. Spaces can now be included in the name of a JRA file, and temporary recordings are always placed in the default `temp` directory (as specified by the system property `java.io.tmpdir`). Note that if you use `jrcmd` on Windows, then a filename that includes spaces must be surrounded by `\"`. For example: `jrcmd 3964 jrarecording time=100 filename=\"name with spaces.jra\"`

Change Request ID	Description
CR301366	Previously, in JRockit Mission Control 2.0, if the startup parameters of a monitored JRockit JVM had changed since the first time a JRockit JVM was monitored, then no longer available attributes were still shown (but without any data) in graphs and other places in the Management console. In JRockit Mission Control 2.0.1, attributes no longer available are greyed out in the Management Console.
CR293482	Before JRockit Mission Control 2.0.1, if the preferences had been changed to run the Memory Leak Detector in a JRockit Mission Control tab on Linux, then when the Trend or Allocation Stack Traces tab was open, the Detector's GUI flickered every time there was an update. This problem has now been fixed in 2.0.1.

Changes in the JRockit Mission Control 2.0 Release

The following has changed in the JRockit Mission Control 2.0 release.

Change Request ID	Description
CR292969	The JRockit Mission Control launcher on Linux, jrmc, has the following command-line options: • `-nl`—sets the locale, for example, `jrmc -nl ja_JP` for Japan and Japanese. The default locale is English. The argument to `-nl`, should follow the same format as return values from `java.util.Locale.toString()`. • `-help`—displays the available options. • `-version`—displays version information of the JRockit JVM and JRockit Mission Control. The JRockit Mission Control launcher on Windows, `jrmc.exe`, has the same options as its Linux counterpart. However, since it uses the "windows" subsystem (and not the "console" subsystem), the output is lost unless it is redirected. If run from a command shell, a way to redirect is to type "jrmc -version |more".
CR293818	If you are running JRockit Mission Control on a Windows system, you need to be a member of the Administrators or the Performance Logs user groups to be able to create a JRA recording. The typical error message, for not being part of either of these groups, can look like this: `[perf ] Failed to init virtual size counter:`

Change Request ID	Description
CR295172	The JRockit Mission Control 2.0 GUI is now available in the Windows **Start** menu.
CR298174	Before, any notification trigger in the Management Console, could be ignored the first time it arrived. This problem has now been fixed.

Known Issues

The following issues are known in the JRockit Mission Control 2.0 and 2.0.1 releases:

Issue	Description
CR355590	In the instance graph of the Memory Leak Detector, false references from a Global JNI Handle might be shown. Any such reference where the tooltip says "Number of handles:1" might be false. Similarly, but less frequently, references from a Threadroot:(MemLeak Socket Reader) are always false. These references are temporary references used in the implementation of the Memory Leak Detector in JRockit Mission Control. They do not keep objects alive.
CR315538	When doing a JRockit Runtime Analyzer recording, one of the pauses in concurrent garbage collections may increase slightly. The pause is normally in the 5 milliseconds range but during the JRA recording, the pause time may increase to around 10 milliseconds. Once the recording is completed, the pause time returns to normal.
CR310655	If you connect the JRockit Memory Leak Detector to the JRockit JVM running the Detector, then the JRockit JVM may run out of memory due to an object allocation feedback loop. You can alleviate this problem by refraining from inspecting stack traces or type graphs of commonly allocated object types, such as `java.lang.String` objects or `char` arrays.
CR313460	If you successively expand a method in the Predecessors panel on the Methods tab in the JRockit Runtime Analyzer, the percent value of the predecessors may start to decrease even if you do not reach a branching point. The percent value should stay the same as long as you do not expand to a branching point. This may happen both in JRockit Mission Control 2.0 and 2.0.1.
CR302355	On Linux, the problem with running the Memory Leak Detector in a tab of the main window with JDK 5.0 has been resolved in JRockit Mission Control 2.0.1. However, with JDK 6, there are other SWT/AWT interoperability problems due to AWT changes. As a consequence, the preference setting for how to run the Memory Leak Detector is now kept separately for each JDK version on Linux, and will only default to run in a tab on JDK 5.0.

Issue	Description
CR307929	The Method Profiler in the JRockit Management Console currently only profiles the first matching class. There is currently no way to profile all matching classes, or a class specific to a certain class loader.
CR311881	On Linux, the value in the **System total #context switches/second** field is 1000 times too low. The field is found in the Threads panel on the General tab of the JRockit Runtime Analyzer. Note that this has been fixed in JRockit Mission Control 2.0.1. **Workaround:** Manually multiply the value presented in the **System total #context switches/second** field by 1000 to get the correct value.
CR293482	If the Memory Leak Detector is run in a JRockit Mission Control tab on Linux with the Trend or Allocation Stack Traces tab open, then the Detector's GUI flickers every time there is an update. Note that this problem has been fixed in JRockit Mission Control 2.0.1. **Workaround:** Always open the Memory Leak Detector in a separate window on Linux. This is the default on affected platforms.
CR289236	In the Management Console, the attributes available to graphs and in other places depend on startup parameters of the monitored JRockit JVMs. If the startup parameters of a monitored JRockit JVM has changed since the first time that JRockit JVM was monitored, attributes that no longer are available may still be shown, but without any data. Other than the missing data, there is no indication that the attributes are inactive. Note that this problem has been fixed in JRockit Mission Control 2.0.1, where no longer available attributes are greyed out.
CR300410	There is a known problem with dark matter analysis in JRA recordings. No dark matter is calculated during the JRA recording. This is a problem on the server side, i.e. not the JRA tool itself. **Workaround:** Set `-XXminblocksize` on the command line of the JRockit JVM to be analyzed, i.e. before you start your application. The value should be set to the minimum TLA size used on that JRockit JVM The default is 2k.
CR301546	When clicking in the JRockit Mission Control MBeanBrowser, a `java.util.ConcurrentModificationException` might be printed in the JRockit Mission Control log file. This should not affect functionality in any way.

Issue	Description
CR302355	Default for JRockit Mission Control 2.0 and Linux is to run the Memory Leak Detector in a separate window, instead of in a tab in the main window. For the JDK 1.4.2 on Linux, this is also the only way to run the Memory Leak Detector.
	For other JDKs, it may be possible to run the Memory Leak Detector in a tab by changing preferences, but GUI updates may cause flickering in various degrees depending on the X server. This is due to SWT/AWT interoperability issues on X11.
	In JRockit Mission Control 2.0.1 on Linux, the Memory Leak Detector can be run flawlessly in a tab with JDK 5.0, but not with JDK 6. The preference settings and defaults have been updated accordingly.
CR302571	There are some limitations on the handling of JRA files, depending how the recordings were started:
	• If started from jrcmd, no spaces are allowed in the given file name or path.
	• If started from JRockit Mission Control on an Oracle JRockit JVM 5.0 or higher, temporary recordings with names starting in jrockit and ending in .jra or .jra.zip are placed in the current working directory of the recording JVM and may be left in that directory even after a successful transfer to JRockit Mission Control.
	Note that these limitations have been removed in JRockit Mission Control 2.0.1.
CR302635	It is possible to create, rename, and delete folder nodes under the **Discovery** folder. Note that this has been fixed in JRockit Mission Control 2.0.1.
	Workaround:
	Do not perform these operations on these nodes. If any of these operations is mistakenly performed on these nodes, you can simply restart JRockit Mission Control and everything will be back to normal.
CR303343	The Compaction Ratio displayed in the Old Collections section on the **GC Events** tab in JRockit Runtime Analyzer (JRA) is not calculated correctly. This happens if the number of available compaction parts is not equal to the default value of 128. Note that this has been fixed in JRockit Mission Control 2.0.1.
	Workaround:
	Calculate the correct value by dividing **# Compacted Parts** by **Available Compaction Parts**.

Issue	Description
CR303920	In JRockit JVM 5.0, attempting to start the management server on a port that is in use, will exit the JVM (with `System.exit(1)`). This is especially dangerous if the management server is started using `jrcmd`, on a JRockit JVM that already is running. Note that this has been fixed in JRockit Mission Control 2.0.1. **Workaround:** Start the management server on JRockit JVM startup with the `-Xmanagement` option (including parameters). or Connect using local attach on the same machine, with JRockit Mission Control.
CR304566	If you are running JRockit Mission Control locally and want to create a JRA recording for a 1.4.2 JRockit JVM on a remote machine, you might run into a "Could not open JRA for `<JVM connection>`" problem. This problem occurs since the JRockit Mission Control GUI itself, only validates local file names when starting a JRA recording; whereas the 1.4.2 JRockit JVM on the remote machine, do not support the automatic transfer of JRA data to your local JRockit Mission Control. This issue has been fixed in JRockit Mission Control 2.0.1. **Workaround:** Use relative file names for your JRA recordings. These file names are interpreted relative to the current working directory of the recording JRockit JVM.
CR304936	When connecting to a JRockit JVM from the Management Browser in JRockit Mission Control, the **Disconnect** button and context menu items are disabled until something else is selected in the Management Browser and the connected JRockit JVM is reselected.
CR305851	The suggested port in the Connection Wizard for a JRockit JVM 1.4.2 is 7091. This is the default port for the JMX communication, but a JRockit JVM 1.4.2 doesn't use the JMX-protocol. This means that the suggested port needs to be 7090, which is the default port for the used RMP protocol. This issue has been fixed in JRockit Mission Control 2.0.1, where the suggested port for a JRockit JVM 1.4.2 is 7090. **Workaround:** Set the port manually in the connection wizard to port 7090.

JRockit Mission Control 1.0 Release Notes

This document contains important details for BEA JRockit Mission Control 1.0. It contains information on the following subjects:

- New Features and Enhancements in BEA JRockit R26.4

- New Features and Enhancements in BEA JRockit R26.3

- New Features and Enhancements in BEA JRockit R26.2

- New Features and Enhancements in BEA JRockit R26.0

- Changes in the BEA JRockit R26.4 Release

- Changes in the BEA JRockit R26.3 Release

- Changes in the BEA JRockit R26.2 Release

- Changes in the BEA JRockit R26.0 Release

- Known Issues

New Features and Enhancements in BEA JRockit R26.4

The main focus on this release has been to stabilize the tools. There are no major features added in this release.

New Features and Enhancements in BEA JRockit R26.3

- Threading model of the JRockit Management Console has been greatly improved.

- Various user interface issues have been fixed in both the JRockit Memory Leak Detector and the JRockit Management Console.

New Features and Enhancements in BEA JRockit R26.2

- JRockit Mission Control is now supported in BEA JRockit 1.4.2 R26.

  Note: Since BEA JRockit 1.4.2 has no platform MBean server, the JRockit Management Console included with this version of BEA JRockit will have less functionality available than the 5.0 version.

New Features and Enhancements in BEA JRockit R26.0

- JRockit Mission Control is now a supported tool set and it contains the following JRockit Mission Control tools:

  – JRockit Management Console

  – JRockit Runtime Analyzer

  – JRockit Memory Leak Detector

- The JRockit Memory Leak Detector and the JRockit Analyzer are now part of the JRockit JDK.

Most Recent Changes

The following has happened with JRockit Mission Control for the different releases:

- Changes in the BEA JRockit R26.4 Release

- Changes in the BEA JRockit R26.3 Release

- Changes in the BEA JRockit R26.2 Release

- Changes in the BEA JRockit R26.0 Release

Changes in the BEA JRockit R26.4 Release

The following has changed in the BEA JRockit R26.4 release.

Change Request ID	Description
CR283027	Due to changes in how persistence work, the historical data view displays persistent attribute data only.
CR273969	Previously the Management Console did not enable persistence without being restarted. This problem has now been fixed.
CR274165	Additional issues in the persistence layer have been fixed: • Persistence data from several connections are now stored in separate files. • If you try to change persistence set while connected, you will be asked to close all your connections, since it is not possible to change persistence set while connected.

Changes in the BEA JRockit R26.3 Release

The following has changed in the BEA JRockit R26.3 release.

Change Request ID	Description
CR208105	An option for changing what attribute is being visualized in a velocimeter has been added to the velocimeter popup-menu in the Management Console.
CR250019	A fix to prevent network delays from freezing the GUI by de coupling GUI functionality from network IO calls has been implemented.
CR261212	The known issue CR261718 has now been fixed.
CR262801	Previously, when trying to start the RMP server on a negative, or already occupied port, it could become impossible to re-start it on a valid port. This has now been fixed.

Changes in the BEA JRockit R26.2 Release

The following has changed in the BEA JRockit R26.2 release.

Change Request ID	Description
CR241626, CR247386	The context sensitive menus in the graphs and in the instance inspector of the Memory Leak Detector are not always disabled even though they are not applicable for use. This problem has now been fixed.
CR244231	The JRockit Runtime Analyzer (JRA) no longer records too many reference objects.
CR250017	The thread stack dump formats for JMAPI and JLMEXT used to differ. Now the JMAPI method `ThreadSystem#getThreadStackDump()` uses the same code path as the JLMEXT `ThreadMXBean#getThreadStackDump()` method. This eliminates confusion resulting from differing dump formats.
CR253583	JRA recordings will force a garbage collection at the start and end of a recording that can sometimes be too intrusive for the application. To disable the forced garbage collections, start the JRA recording with the option `heapstats=false`. When disabled, some information will be missing from the JRA recording.
CR254819	Now the port can be specified for both the `start_management_server` and the `start_rmp_server` control break handlers. Use: `start_management_server port=<port#>`.
CR254636, CR255962	When sampling allocation intensive applications with the JRA, it can cause an unexpectedly high performance impact. This problem has now been fixed.

Changes in the BEA JRockit R26.0 Release

The following has changed in the BEA JRockit R26.0 release.

Change Request ID	Description
CR180667	A list with server names that are discovered through JDP now supplements the server name entry field in the "New Connection" dialog box.
CR206186	Chart views in the JRockit Management Console now remember their attribute subscriptions when the application is shut down and restarted.
CR208294	Previously only public methods were displayed in the method selection dialog for the method profiler. Now all declared methods are visible.

Change Request ID	Description
CR210028	Previously native/abstract methods, which cannot be profiled, were displayed in the method selector for the method profiler. Now such method are not displayed.
CR210030	Methods that have already been selected for profiling in the method profiler are greyed out in the **Add Method** wizard, which means that they cannot be selected.
CR210518	The communications preferences have been changed to include the default update time only. It is also persisted in the settings file.
CR211708, CR238905	The JRockit Runtime Analyzer now requires a license. There are two types of licenses: a free developer license and a license for use in production. For more information, please see: http://dev2dev.bea.com/jrockit/tools.html
CR222700	A check box that makes it possible to block automatically discovered JRockit JVM instances on the network has been added to the **Preferences** dialog in the JRockit Management Console.
CR228592	Using the Memory Leak Tool could in some instances make the JVM freeze or crash. This has been fixed.
CR231402	Previously, MBeans without any attributes were not shown in the attribute inspector of the JRockit Management Console. Now all MBeans are shown.
CR237249	When starting a Memory Leak Detector session, the Memory Leak Detector first connects to the Management Server to start up the native Memory Leak Server. The client subsequently connects to the Memory Leak Server using an anonymous port. If you have tight firewall demands, you can now specify what listening port the native Memory Leak Server should use by specifying the system property `com.jrockit.memleak.mlsport` on the command line when starting the Memory Leak Detector.
CR238194	The maximum capacity of the data buffer for non-persistent graph data (in the JRockit Management Console) now dynamically matches the free heap space.
CR238196	The attribute inspector in the JRockit Management Console can now show attributes of the type `TabularData`.
CR238205	Synthetic attributes for the live set and the garbage collection pause times have been added in the JRockit Management Console.
CR238769	In the JRockit Management Console persisting the attribute data now works as intended.

Change Request ID	Description
CR238808	A log rotation setting is now available through the **Persistence** tab in the Preferences dialog of JRockit Management Console.
CR240694	The auto y-range setting on attribute subscriptions in the JRockit Management Console now uses the scaled values to determine the range of the graph axis.
CR241880	A refresh button to refresh the attribute data for the MBeans that are shown in the JRockit Management Console has been added.
CR243090	A notification action that writes thread stack dumps to an application alert, a log file, or both, when notified by a trigger (such as high CPU usage) has been added to the JRockit Management Console.
CR243386	A button has been added to the method profiler, in the JRockit Management Console, where you can select and deselect all methods in a class.
CR245479	Previously, the filtering of methods did not work properly in the JRockit Runtime Analyzer (JRA). When the filtering was set, the resulting method table could unexpectedly return blank. Now this has been fixed.
CR246870	The JRA counters for "number of exceptions thrown" and "number of freelist misses" during a recording were incorrect if several consecutive JRA recordings were performed in the same JVM instance. The counters are now reset before each new recording.
CR247601	This release of BEA JRockit includes the Mercury Diagnostics Profiler. This free profiler provides a way for Java/J2EE development teams to profile applications in the development environment. For more information, please see: http://www.mercury.com/us/products/diagnostics

Known Issues

The following issues are known in the BEA JRockit R26 releases:

Issue	Description
CR274165	There are some known issues with the persistence layer in the Management Console. • Changing the persistence set while connected, requires that you disconnect from all connected JVM instances and clear the gathered data. • Changing the persistence set while viewing a historical data graph, can result in the persisted data not being properly cleared. If the data was not properly cleared, an exception may be thrown the next time you start collecting new persistent data.
CR241626, CR247386	The context sensitive menus in the graphs and in the instance inspector of the Memory Leak Detector are not always disabled even though they are not applicable for use. If you select such an invalid operation, you will normally receive an error message. **Note:** This issue has been fixed in BEA JRockit R26.2.
CR244231	The JRockit Runtime Analyzer (JRA) records too many reference objects. JRA recordings sample how many reference objects that are live and have a referent. This number is an upper bound of how many they actually are, i.e. there are not more than the counted number but might, for different garbage collection configurations, be fewer. This is applicable for soft, weak, and phantom references. **Workaround:** There is currently no workaround. **Note:** This issue has been fixed in BEA JRockit R26.2.
CR249007, CR249235	A minor issue concerning leaking memory has been discovered in the JRockit Runtime Analyzer. The leak will only become significant when a large number of recordings are created over time. **Note:** This issue has been fixed in BEA JRockit R26.2.
CR250954	When you right-click on a method in the JRA client, you can choose to **Launch external program**. The intended functionality (to enable integration with an external source code browser) is currently not enabled.

Issue	Description
CR254636, CR255962	When sampling allocation intensive applications with the JRA, it can cause an unexpectedly high performance impact.
	Note: This issue has been fixed in BEA JRockit R26.2.
CR261718	Running the Memory Leak Detector with different locales may corrupt the preferences setting for "Lowest heap usage to report". This setting can become 100% or greater, showing no types or 0%, showing all types.
	Workaround: Edit the preference setting to any accepted value (e.g. "0.001", but in your current locale). Or, while the tool is not running, delete the settings file `~/.MemoryLeakDetector/memleak.xml`.
	Note: This issue has been fixed in BEA JRockit R26.3.