



BEA WebLogic JRocket™ 7.0 SDK

User Guide

Release 7.0 Service Pack 5
March 2004

Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic JRockit, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Contents

Introduction

Installing JRockit.....	1-1
Hardware and Operating System Requirements.....	1-2
Tuning JRockit	1-2
Reporting Problems and Bugs	1-2
If WebLogic JRockit Crashes.....	1-3

Using BEA WebLogic JRockit 7.0 JVM Command Line Options

Using Standard Command Line Options.....	2-1
Using the -X Options.....	2-2

Using the WebLogic JRockit 7.0 JVM Management Console

Console Overhead	3-2
Starting the Console	3-2
Enable the Management Server.....	3-2
Start the Management Console	3-3
Set the Port	3-3
Change the Number of Connections	3-3
Parts of the Console.....	3-4
Setting Up the Console.....	3-6
Making Connections	3-6
Creating a New Folder	3-6
Creating a New Connection	3-7
Connecting a Connection to WebLogic JRockit 7.0 JVM.....	3-8
Disconnecting a Connection from WebLogic JRockit 7.0 JVM	3-8
Renaming a Connection or Folder	3-9
Removing a Connection or Folder.....	3-10

Hiding Disconnected Connections	3-10
Enabling Console Settings	3-10
Setting the Operation Mode	3-11
Setting Other Preferences	3-12
Customizing the Display	3-14
Using the Settings File	3-16
Using the Console	3-17
Information Tabs	3-17
Overview Tab	3-17
Memory Tab	3-19
Processor Tab	3-20
System Tab	3-22
Notification Tab	3-23
View Historical Data	3-31
Using Advanced Features of the Console	3-33
View Thread Stack Dump	3-33
Method Profiling Tab	3-34
Exception Counting Tab	3-37
Closing the Console	3-39

1 Introduction

BEA WebLogic JRockit 7.0 SDK provides tools, utilities, and a complete runtime environment for developing and running applications using the Java programming language. The WebLogic JRockit SDK includes the WebLogic JRockit Java Virtual Machine (JVM), the first commercial server-side JVM. The WebLogic JRockit JVM was developed uniquely for server-side applications and optimized for Intel architectures to ensure reliability, scalability, and manageability for Java applications.

This user guide provides instructions for using WebLogic JRockit on the Window and Linux platforms.

This section, [Introduction](#), contains information on the following subjects:

- [Installing JRockit](#)
- [Hardware and Operating System Requirements](#)
- [Tuning JRockit](#)
- [Reporting Problems and Bugs](#)
- [Reporting Problems and Bugs](#)

Installing JRockit

For installation instructions, please see the [BEA WebLogic JRockit 7.0 SDK Installation and Migration Guide](#).

Hardware and Operating System Requirements

BEA WebLogic JRockit 7.0 SDK runs on 32-bit Intel Pentium (Pentium Pro or better) and Xeon systems as well as on 64-bit Intel Itanium Platform Family systems. For a complete listing of the hardware and O/S configurations for BEA JRockit 7.0 (J2SE 1.3.1), please refer to [BEA JRockit 7.0 \(J2SE 1.3.1\) Supported Configurations](#) at:

http://edocs.bea.com/jrockit/jrdocs/suppPlat/supp_70.html

Tuning JRockit

For information on tuning WebLogic JRockit's threading and memory management features, please refer to Performance Tuning at:

<http://edocs.bea.com/wljrockit/docs70/tuning/index.html>

Reporting Problems and Bugs

Should you experience any problems or find any bugs, please send us an e-mail at support@bea.com. We would appreciate if you could provide as much information as possible about the bug such as hardware, version, the program you ran, stack dumps (if any), and a small code example that will reproduce the bug.

If WebLogic JRockit Crashes

If JRockit crashes, it will dump the information to `stderr`. It will also create, in the directory where the VM was started, a file called `jrockit.dump`, which contains the same information as the dump to `stderr`.

1 *Introduction*

2 Using BEA WebLogic JRockit 7.0 JVM Command Line Options

This section describes how to use WebLogic JRockit 7.0 JVM's command line options. WebLogic JRockit supports all the standard Java command line options plus numerous extended options.

This section includes information on the following subjects:

- [Using Standard Command Line Options](#)
- [Using the -X Options](#)

Note: If WebLogic JRockit 7.0 JVM behaves in some unexpected way, please consult the WebLogic JRockit 7.0 SDK FAQ. If that doesn't solve your problem, please send an e-mail to support@bea.com

Using Standard Command Line Options

The following are the standard command line options you can use with WebLogic JRockit 7.0 JVM:

- `-classpath <directories and zips/jars separated by : or ;>`
Tells WebLogic JRockit 7.0 JVM where to look for classes and resources.

Use a semicolon to separate entries on Windows systems and a colon on Linux.

- `-cp <directories and zips/jars separated by : or ;>`

This is short for `-classpath`.

- `-D<name> [=<value>]`

Tells WebLogic JRockit 7.0 JVM to set a Java system property. These can be read by a Java program, using the methods in `java.lang.System`.

- `-version`

Tells WebLogic JRockit 7.0 JVM to display its product version number and then exit.

- `-showversion`

Tells WebLogic JRockit 7.0 JVM to display its product version number and then continue.

- `-verbose[:<components separated by ,>]`

Tells WebLogic JRockit 7.0 JVM to display verbose output. This option is used mainly for debugging purposes and causes a lot of output to the console. Supported components are `memory`, `load` and `codegen`. If no component is given, WebLogic JRockit 7.0 JVM will display verbose information on everything.

- `-help`

Tells WebLogic JRockit 7.0 JVM to display a short help message.

- `-X`

Tells WebLogic JRockit 7.0 JVM to display a short help message on the extended options.

Using the -X Options

The `-X` command line options are used to change the behavior of WebLogic JRockit 7.0 JVM to better suit the needs of different Java applications.

- `-Xbootclasspath:<directories and zip/jar files separated by ; or :>`

Tells WebLogic JRockit 7.0 JVM where to look for the Java 2 class libraries. Use this option only if you want to run WebLogic JRockit with class libraries other than those provided with the default installation. Use semicolons to separate entries on Windows systems and colons to separate entries on Linux.

- `-Xms<size> [m|M|k|K]`

Specifies the initial heap size in megabytes [m|M] or kilobytes [k|K].

- `-Xmx<size> [m|M|k|K]`

Specifies the maximum heap size in megabytes [m|M] or kilobytes [k|K].

Default is 75% of physical memory.

- `-Xss<size> [k|K]`

Sets the thread stack size in kilobytes.

- `-Xgc: [gencopy|gencon|singlecon|parallel]`

Specifies which type of garbage collector to use:

<code>gencopy</code>	Generational Copying
<code>gencon</code>	Generational Concurrent
<code>singlecon</code>	Single Spaced Concurrent
<code>parallel</code>	Parallel

Default is `gencopy` if `-Xmx` is set to less than 128M and `gencon` if it is set to more.

- `-Xns<size> [m|M|L|K]`

Sets the initial nursery size for generational collectors.

- `-Xallocationtype: <global|local>`

Sets the type of allocation to perform.

- Global—all threads allocate from the same heap. This is the default for the `gencopy` garbage collection method.
- Local—each thread allocates from its own heap-area. Local allocation is faster if you know that the number of threads is low. This is the default for the `singlecon`, `gencon` and `parallel` garbage collection methods.

- `-Xcleartype:<gc|local|alloc>`

Clears memory during garbage collection, either when a new thread local area is allocated or when an object is allocated, `alloc` is not available on IA64.
- `-Xnoclassgc`

Switches off garbage collection of classes.
- `-Xgcreport`

Writes an extensive memory management report at the end of a run.
- `-Xgcpause`

Prints pause times caused by the garbage collector.
- `-Xnoopt`

Tells WebLogic JRockit not to optimize code.
- `-Xnativethreads`

Tells WebLogic JRockit to use operating system threads instead of the WebLogic JRockit 7.0 JVM high performance threading system (thin threads). The default, if neither `-Xnativethreads` nor `-Xthinthreads` is specified, is to use `-Xnativethreads`.
- `-Xthinthreads`

Tells WebLogic JRockit to use the WebLogic JRockit 7.0 JVM high performance threading system instead of the operating system threads. This option is recommended when running applications with a large number of threads (more than several hundred).

Note: `-Xthinthreads` is not available on IA64.
- `-Xverify`

Do complete bytecode verification.
- `-Xquiet`

Tells WebLogic JRockit not to dump any informational output to `stdout` or `stderr`.

DISCLAIMER: The `-x` options are non-standard and subject to change without notice.

3 Using the WebLogic JRockit 7.0 JVM Management Console

The Management Console can be used to monitor and control running instances of WebLogic JRockit 7.0 JVM. It provides real-time information about the running application's characteristics, which can be used both during development—for example, to find where in an application's life cycle it consumes more memory—and in a deployed environment—for example, to monitor the system health of a running application server.

This section includes information on the following subjects:

- [Console Overhead](#)
- [Starting the Console](#)
- [Parts of the Console](#)
- [Setting Up the Console](#)
- [Using the Console](#)

Console Overhead

The extra cost of running the Management Console against a running WebLogic JRockit 7.0 JVM is very small and can almost be disregarded. This provides for a very low cost monitoring and profiling of your application.

Note: You should not run the Management Console on the same machine as the VM you are monitoring. Doing this will degrade performance significantly.

Starting the Console

Starting the Management Console is a two-step process:

1. [Enable the Management Server](#)
2. [Start the Management Console](#)

Additionally, you might want to also complete these tasks as part of the start-up process:

- [Set the Port](#)
- [Change the Number of Connections](#)

Enable the Management Server

Before the Management Console can connect to WebLogic JRockit 7.0 JVM, the management server in the VM needs to be started. The server is disabled by default. To enable the management server, start WebLogic JRockit 7.0 JVM set the `-Xmanagement` option:

```
-Xmanagement
```

Start the Management Console

Start the Management Console from the command prompt by typing:

```
console
```

Note: Before starting the Management Console, you must specify the JRE path and the classpath to the `.jar` file.

Set the Port

When WebLogic JRockit 7.0 JVM is started with the `-xmanagement` option set—and provided the VM is not running in “quiet” mode—it should print out a short message following the command line indicating that the management server is running and which port it is using. You can optionally choose which port to use by setting the port in the `port` property:

```
-Djrockit.managementserver.port=<portnumber>
```

The default port the management server uses to connect is 7090. It is strongly recommended that you block this port in your firewall, otherwise unauthorized users might access the management server.

Change the Number of Connections

You can change the number of connections allowed to the server by setting the `maxconnect` property:

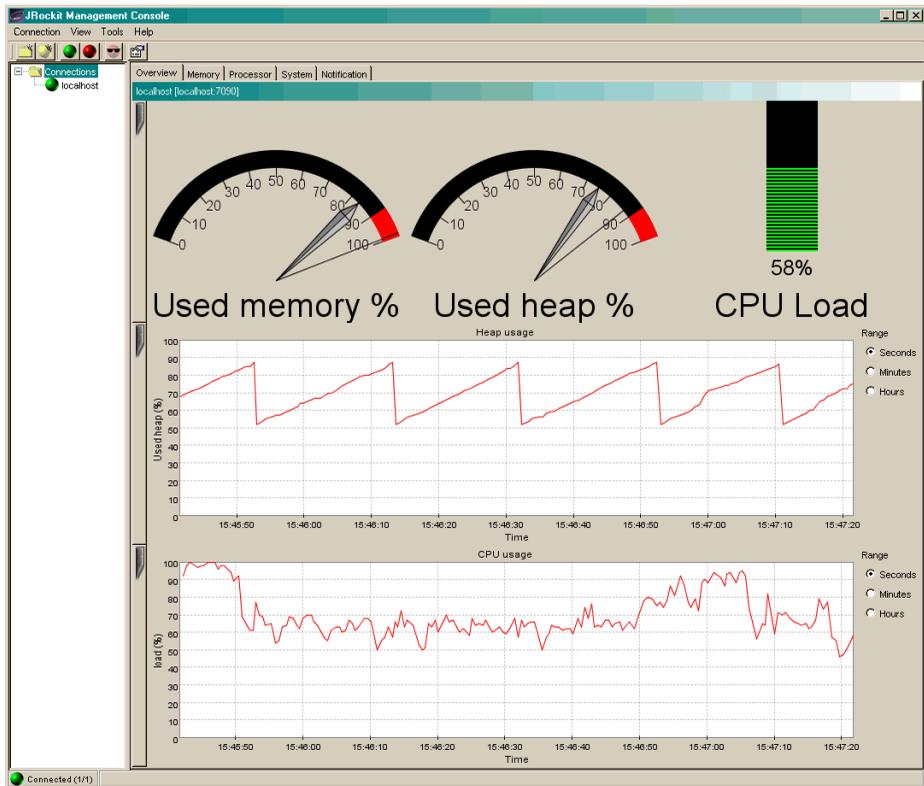
```
-Djrockit.managementserver.maxconnect=<maximum number of connections>
```

The default limit is four concurrent connections. While this should be enough for most users, you can change it, if necessary. The connection limit protects against Denial of Service (DoS) attacks by intruders.

Parts of the Console

When the Management Console window appears, the console has started, as shown in [Figure 3-1](#):

Figure 3-1 WebLogic JRockit 7.0 JVM Management Console



The Management Console window is divided into two panes: a connection browser tree in the left pane ([Figure 3-2](#)) and a tabbed interface in the right pane ([Figure 3-3](#)).

Figure 3-2 Connection Browser**Figure 3-3 Information Tabs (Administrator Mode)**

The first tab shows an Overview of information for the selected WebLogic JRockit 7.0 JVM connection(s) (as highlighted in the connection browser pane). The other tabs contain detailed information about different areas of the VM, as will be described in [“Information Tabs”](#) on page 17.

[Figure 3-3](#) shows the information tabs available in the console’s Administrator operation mode. When the console is in the Developer mode, additional tabs appear, as shown in [Figure 3-4](#). These two operation modes are described in [“Setting the Operation Mode”](#) on page 11.

Figure 3-4 Information Tabs (Developer Mode)

The console includes a toolbar that contains command buttons for some of the menu options ([Figure 3-5](#)). To toggle the Toolbar on or off, on the View menu select Tool Bar.

Figure 3-5 Management Console Toolbar

The status bar (Figure 3-6) at the bottom of the window displays informational messages and tool tips when you hover over a toolbar button or select something in a menu. It also indicates whether the Management Console is connected to one or several WebLogic JRockit 7.0 JVM implementations or not. To toggle the Status Bar on or off, on the View menu, select Status Bar.

Figure 3-6 Status Bar



Setting Up the Console

Once the console is running, you will need to configure it to suit your needs. Configuring—or “setting up”—the console includes these tasks:

- [Making Connections](#)
- [Enabling Console Settings](#)

Making Connections

The connection browser displays a collection of saved connections to WebLogic JRockit 7.0 JVM organized in folders. If necessary, you can add your own folders and connection nodes to the tree structure. Active connections currently connected to a running VM are indicated by a green icon; those disconnected are indicated by a red icon.

Creating a New Folder

To create your own folder in the connection browser, do the following:

1. Select an existing folder (for example, Connections) for which you want to create a subfolder.
2. Open the New Folder dialog box by doing one of the following:

- Choose Connection→New Folder.
- Press the right mouse button to open a context menu and select New Folder.
- Press `Ctrl+N`.
- Click the New Folder button on the toolbar.

The Add new folder dialog box (Figure 3-7) appears:

Figure 3-7 Add New Folder Dialog Box



3. Enter the name of the new folder in the text field and click OK.

The new folder will appear in the connection browser.

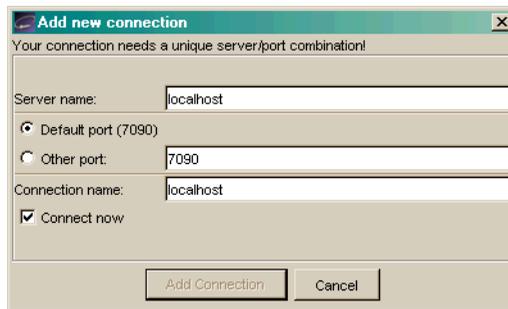
Creating a New Connection

To create a new connection to WebLogic JRockit 7.0 JVM in the connection browser, do the following:

1. Select the folder in which the connection should be placed
2. Open the New Connection dialog box by doing one of the following:
 - Open the Connection menu and select New Connection.
 - Press the right mouse button to open a context menu and select New Connection.
 - Click the New Connection button on the toolbar.

The Add new connection dialog box (Figure 3-8) appears:

Figure 3-8 Add New Connection Dialog Box



3. Enter the name of the server, the port and the new connection in the appropriate text fields or retain the default values. Then, select or deselect Connect now and click Add Connection.

Connecting a Connection to WebLogic JRockit 7.0 JVM

To connect to WebLogic JRockit, do the following:

1. Select the WebLogic JRockit 7.0 JVM connection to connect, a subfolder of connections to connect, or the folder Connections to connect all existing connections.
2. Do one of the following to connect the selected connection(s):
 - Open the Connection menu and select Connect.
 - Press the right mouse button to open a context menu and select Connect.
 - Press `Ctrl+O`.
 - Click the Connect button on the toolbar.

When the connection is made, the status bar will read “Connected” and activity on the console will commence.

Disconnecting a Connection from WebLogic JRockit 7.0 JVM

To disconnect a connection from WebLogic JRockit 7.0 JVM, do the following:

1. Select the WebLogic JRockit 7.0 JVM connection to connect, a subfolder of connections to connect, or the folder Connections to disconnect all existing connections.
2. Do one of the following to disconnect the selected connection(s):
 - Open the Connection menu and select Disconnect.
 - Press the right mouse button to open a context menu and select Disconnect.
 - Press `Ctrl+D`.
 - Click the Disconnect button on the toolbar.

The connection will be lost and the status bar will indicate that you've been disconnected. All activity on the console will cease.

Renaming a Connection or Folder

To rename a connection or a folder of connection, do the following:

1. Select the WebLogic JRockit 7.0 JVM connection or folder to rename.
2. Do one of the following to rename the selected connection or folder:
 - Open the Connection menu and select Rename.
 - Press the right mouse button to open a context menu and select Properties.
 - Press `F2`.
 - Click the name label of the item (see **Note**, below).

The Folder properties dialog box ([Figure 3-9](#)) appears:

Figure 3-9 Folder Properties Dialog Box



3. Enter a new name into the text field and click OK

Note: If you select the last option (click the item label), the Folder properties dialog box will not appear. Instead, the label itself will be enabled for direct editing. Simply type the new name over the old and click away from the label or press Enter.

Removing a Connection or Folder

To remove a connection or folder, do the following:

1. Select a connection or a subfolder to remove.
2. Do one of the following to remove the selected item:
 - Open the Connection menu and select Remove.
 - Press the right mouse button to open a context menu and select Remove.
 - Press `Delete`.
3. Click Yes on the confirmation dialog box that appears.

The selected item disappears from the connection browser.

Hiding Disconnected Connections

Sometimes you might want to show just information about active WebLogic JRockit 7.0 JVM connections. To hide information about disconnected connections, do one of the following:

- Open the View menu and select Hide Disconnected.
- Click the Hide Disconnected button on the toolbar.

To show the information about disconnected connections again, simply deselect Hide Disconnected in same way that you made the selection.

Enabling Console Settings

This section describes how to enable various Management Console settings.

Setting the Operation Mode

The Management Console can be run in two different operating modes:

- **Administrator Mode;** This is the default mode, designed for system administrators who are interested in observing the state of the WebLogic JRockit 7.0 JVM.
- **Developer Mode;** The developer mode is for developers and provides additional features such as a rudimentary method profiler and exception count functionality. Additional pages appearing in the developer mode are the Method Profiler page and the Exception Count page.

To set the operation mode, do the following:

1. From the Tools menu, select Preferences...

The Preferences dialog box (Figure 3-10) appears:

Figure 3-10 Preferences Menu (General Tab)



2. Click the Mode of operation drop-down control to display the list of operation modes (Figure 3-11).

Figure 3-11 List of Operation Modes



3. Select the mode you want to use and click OK.

Depending upon the mode to which you are toggling, the tabs on the console will change. See [Figure 3-3](#) and [Figure 3-4](#) for examples.

Setting Other Preferences

In addition to setting the operation mode, you can use the Preferences dialog box to change these settings:

- Default e-mail settings for the notification system.
- Persistence behavior.

To change either of these values, open the Preferences dialog box from the Tools menu and proceed as described in the following sections:

Setting E-mail Preferences

To change e-mail preferences, do the following:

1. Display the General tab on the Preferences dialog box
2. In the appropriate text fields, enter the new e-mail information (SMTP server and E-mail address), as shown in [Figure 3-12](#).
3. Click OK

Figure 3-12 E-mail Preferences Panel



Enabling Persistence

Enabling the persistence means that aspect values are saved to a file and can be reviewed in charts by opening the View menu and selecting View Historical Data (“View Historical Data” on page 31).

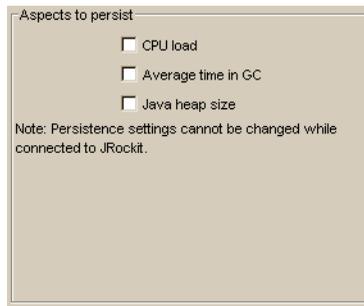
Selecting Aspects to Persist To set persistence preferences, do the following:

1. Disconnect any WebLogic JRockit 7.0 JVM connections.

Note: If you have not disconnected the connections and attempt to use this dialog box, you will be prompted to disconnect.

The checkboxes in the Aspects to persist panel become enabled (Figure 3-13):

Figure 3-13 Aspects to Persist Panel



2. Select the aspects you want to persist.
3. Click OK.

The selected aspect values are saved to a file that you can review in charts as described in “View Historical Data” on page 31.

Specifying the Persistence Directory In addition to setting preferences for the aspects to persist, you can also specify where to save the file that contains the aspect value (the “Persistence directory”). To do so:

1. Click Choose (next to the Persistence directory field).

If you are still connected to WebLogic JRockit 7.0 JVM, you will be prompted to disconnect; click Yes to proceed. A standard Open dialog box appears.

2. Locate the directory where you want to save the file and click Open.

3 Using the WebLogic JRockit 7.0 JVM Management Console

The Open dialog box closes, returning you to the Preferences dialog box.

3. Click OK.

The new Persistence directory will appear in that field.

Erasing Persistence Value Logs Finally, you can erase all persistence value logs by clicking Clear all aspect logs. You will see a confirmation message to which you should respond Yes.

Customizing the Display

You can customize the console and change the way some of the monitoring data is displayed, as described in this section.

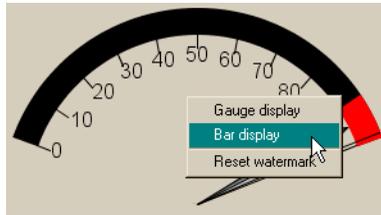
Customizing Gauges and Bars

The gauges and bars are graphical devices showing memory and processor usage ([Figure 3-14](#)).

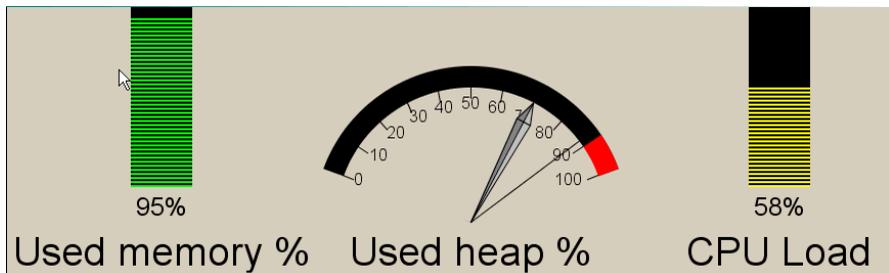
Figure 3-14 Gauges and Bars



- To change from a gauge display to bar display, press the right mouse button when pointing at the gauge and select Bar display, as shown in [Figure 3-15](#).

Figure 3-15 Gauge Context Menu (Bar Display Selected)

The selected gauge will appear as a bar (Figure 3-16).

Figure 3-16 Gauges and Bars with Gauge Converted to a Bar Display

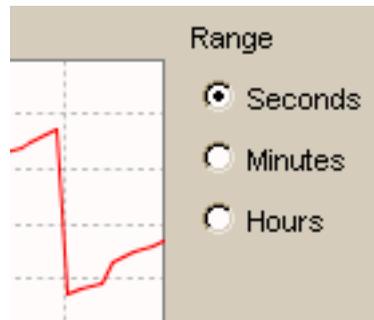
- To change back to a gauge, repeat the above, but select Gauge display.
- To reset the watermark—which indicates the highest level measured so far—press the right mouse button when pointing at the gauge or bar and select Reset Watermark.

Customizing Charts

Charts appear on the Management Console to show specified information about WebLogic JRockit.

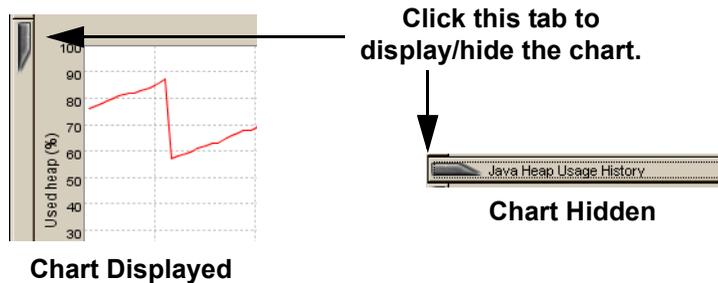
- To change scale on any of the chart, select the desired scale unit (seconds, minutes or hours) to the right of the chart (Figure 3-17) to be changed.

Figure 3-17 Range Selection Radio Buttons



- To hide a chart click the vertical tab at the left of the diagram you want to hide. When the diagram is hidden, the tab appears horizontally (Figure 3-18).

Figure 3-18 Hiding a Chart



- To show the diagram again, click the horizontal tab again.

Using the Settings File

When you exit the Management Console, your settings are automatically saved in a file called `consolesettings.xml`. This file is located in the folder:

```
<user home directory>\ManagementConsole
```

The exact path to the user home directory will vary on different platforms. On Windows it is usually something like `\Documents and Settings\<username>`; for example:

```
C:\Documents and Settings\jsmith\ManagementConsole
```

If no settings file exists in this directory it will be automatically created the next time the Management Console is closed.

Warning: Do not edit this file by hand! Doing so can make it unusable and may cause the Management Console to crash on startup.

If you are experiencing problems with the settings file, you can always delete it and let the Management Console create a new one for you.

Using the Console

The Management Console monitors different “aspects” of WebLogic JRockit 7.0 JVM. An aspect is data that can be measured; for example, used heap size or VM uptime.

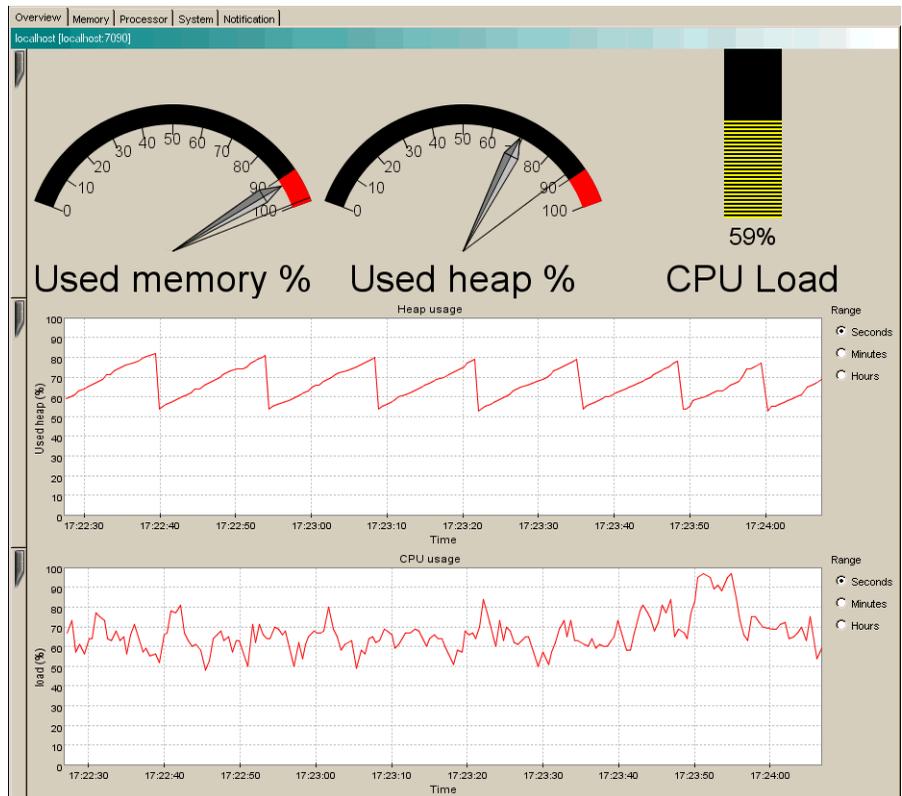
Information Tabs

Information tabs are pages containing details about different areas of the monitored WebLogic JRockit 7.0 JVM. Display a tab by clicking it or by accessing the View menu. This section describes the tabs available on the Management Console.

Overview Tab

The Overview tab ([Figure 3-19](#)) shows an overview of selected connections. To select more than one connection, select the folder containing the connections you want to view. They will appear simultaneously. The page is divided into a “dashboard” with gauges in the upper part and charts in the lower part.

Figure 3-19 Overview Tab

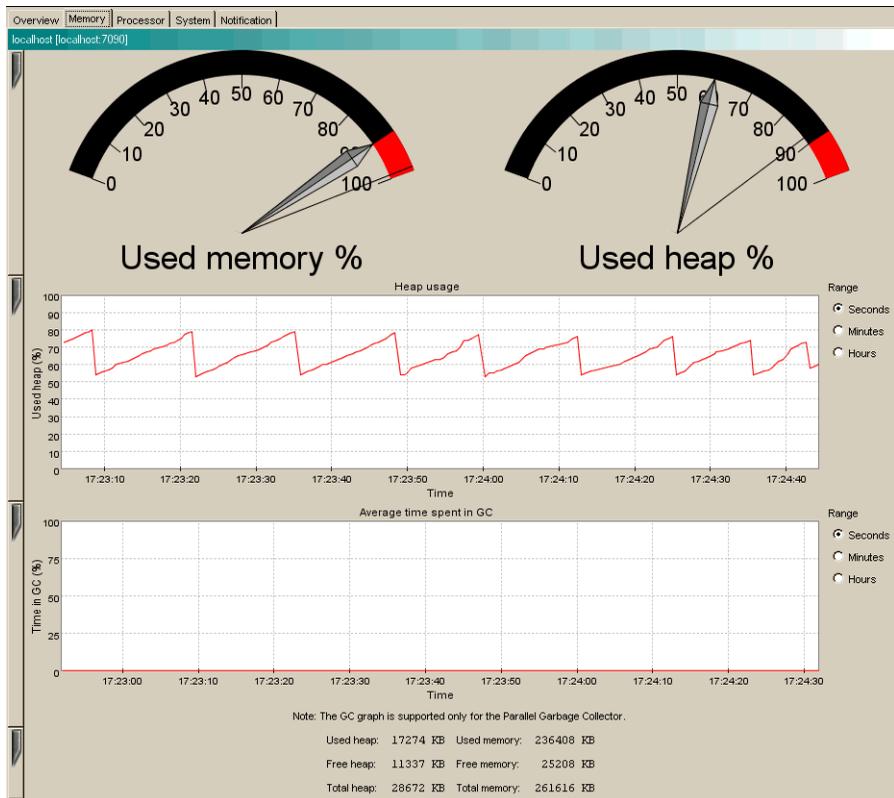


- The **Used Memory** gauge shows the percentage of occupied physical memory on the computer.
- The **Used Heap** gauge shows the percentage of occupied Java heap memory in the VM.
- The **CPU Load** bar shows the usage rate of the processor - or the average processor load on a multi-processor machine.
- The **Heap Usage** chart shows the percentage of used Java heap over time.
- The **CPU Usage** chart shows the average usage rate of the processor(s) over time.

Memory Tab

The Memory tab (Figure 3-20) shows information about the memory status of the system, as shown.

Figure 3-20 Memory Tab



- The **Used Memory** gauge shows the percentage of machine memory in use.
- The **Used Heap** gauge shows the percentage of occupied Java heap.
- The **Heap Usage** chart shows the percentage of occupied heap over time.
- The **Time in GC** chart shows the average time spent on garbage collection over time. This chart is only updated when running WebLogic JRockit 7.0 JVM with the Parallel garbage collector, and an actual garbage collection occurs.

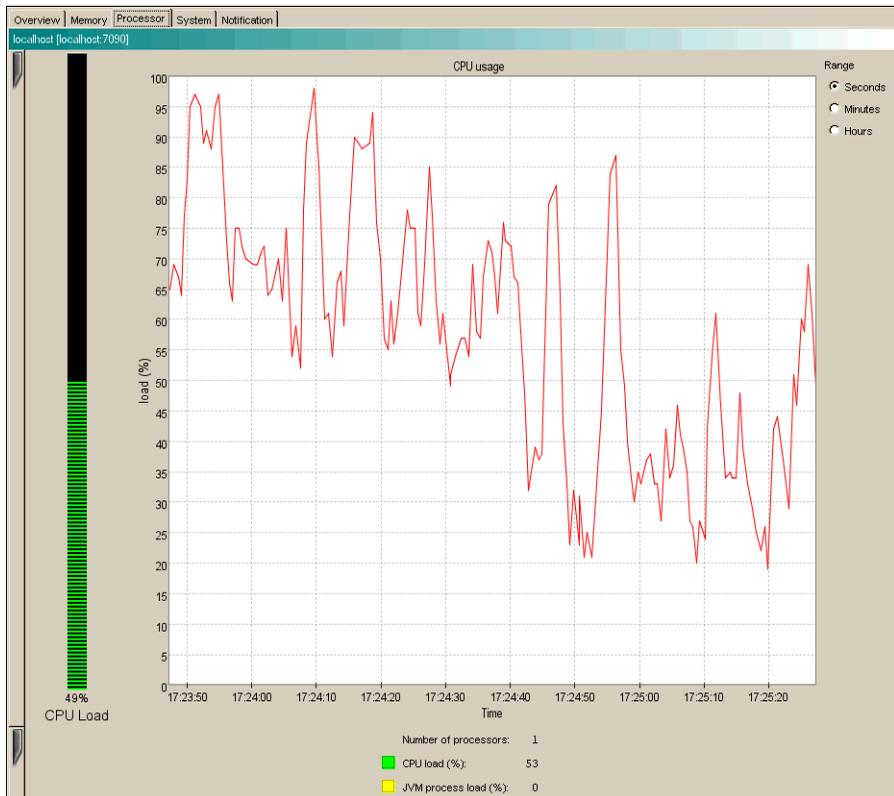
At the bottom of the page the following text information is displayed (in kilobytes):

- **Used Heap** shows the occupied heap space.
- **Free Heap** shows the free heap space.
- **Total Heap** shows the heap size.
- **Used Memory** shows the amount of occupied physical memory.
- **Free Memory** shows the amount of free physical memory.
- **Total Memory** shows the total physical memory size.

Processor Tab

The Processor tab ([Figure 3-21](#)) shows information about the processor status of the system.

Figure 3-21 Processor Tab



- The **CPU Load** bar shows the average processor load as a percentage. The overall load is displayed in green while the load of the JVM process(es) is displayed in yellow.
- The **CPU Usage** chart shows the average processor load as a percentage over time.

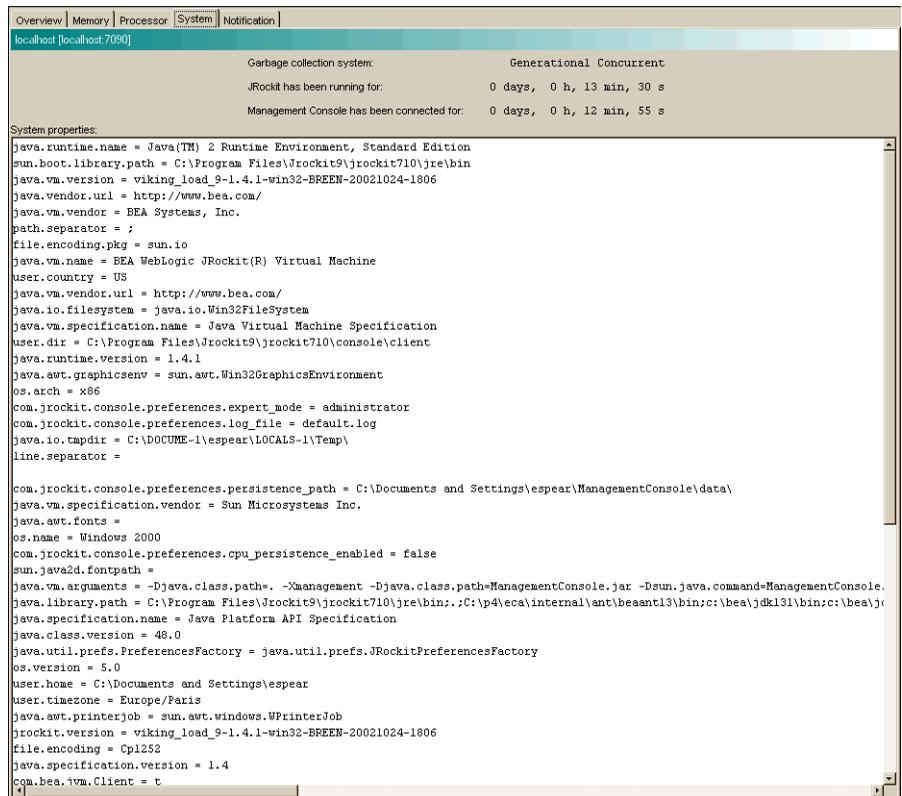
At the bottom of the page the following text information is displayed:

- **Number of Processors** shows the number of processors.
- **CPU Load** shows the overall processor load as a percentage.
- **JVM Process Load** shows the load of the WebLogic JRockit 7.0 JVM process(es), expressed as a percentage.

System Tab

The System tab (Figure 3-22) shows various information about the system status.

Figure 3-22 System Tab



- **Garbage Collection System** shows which garbage collector WebLogic JRockit 7.0 JVM is running.
- **JRockit Uptime** shows how long WebLogic JRockit 7.0 JVM has been running.
- **Connection Uptime** shows how long the currently displayed connection has been connected.
- **Process Affinity** contains buttons that correspond to processors. It displays a green icon if WebLogic JRockit 7.0 JVM is running on this processor and a red

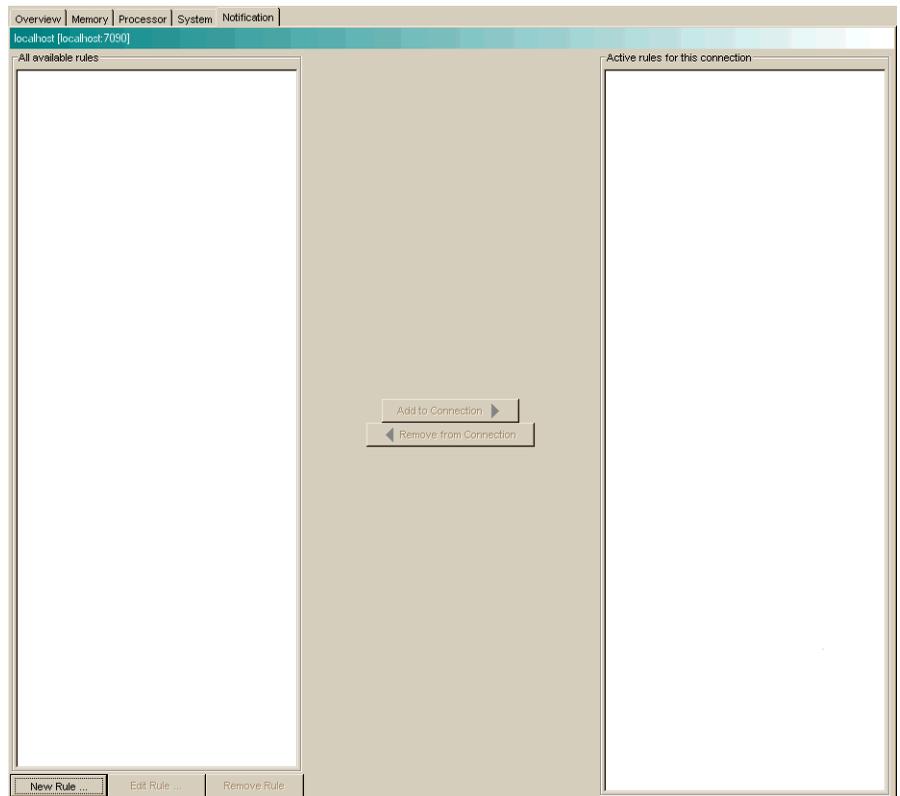
icon if it is not. By selecting a button, the WebLogic JRockit 7.0 JVM process can be bound to one or more processors. The VM might be released from such a connection by deselecting the button again. This is only a suggested affinity: the operating system might not follow the suggestion. Changing the process affinity is a feature that is only available when monitoring a the VM instance running on the Windows platform. The Process Affinity display is only activated when the Management Console is in the Developer mode, described in [“Setting the Operation Mode” on page 11](#).

- **System Properties** shows the Java System Properties loaded in the VM.

Notification Tab

Use the Notification tab ([Figure 3-23](#)) to define alerts that notify users when certain events occur. You can create your own notification rules based on different triggers, with optional constraints, that alert you with a prescribed notification. This section describes how to create these rules.

Figure 3-23 Notification Tab (No Rules Defined)



A notification trigger can be a certain event, for example, that the connection to WebLogic JRockit 7.0 JVM was lost, or that an aspect reaches a certain value, for example, the used memory reaches 95%. A notification constraint can limit when a rule is triggered for example by not sending alerts at night or on certain dates.

The notification action is how the alert is communicated to the user. It can be one of the following:

- E-mail shows an e-mail with the notification is sent to the specified address by using the specified SMTP server.
- System out action displays the notification in the command window where you started the Management Console.

- Application alert displays the notification in an alert dialog in the Management Console.
- Log to file logs the notification to the specified file.

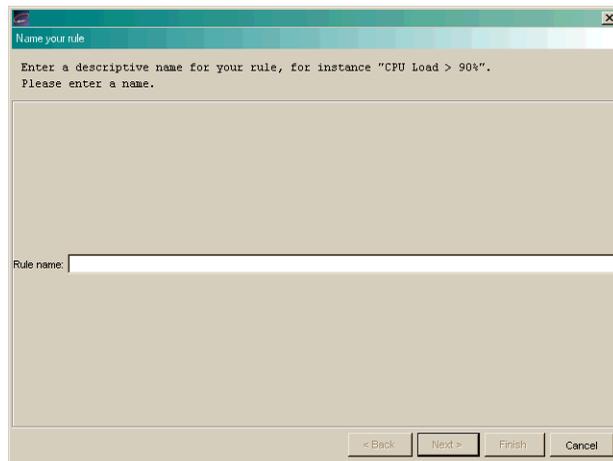
Creating a New Rule

Rules determine when and how to issue a notification. To create a new rule, do the following:

1. Click New Rule.

The Name rule dialog box appears (Figure 3-24):

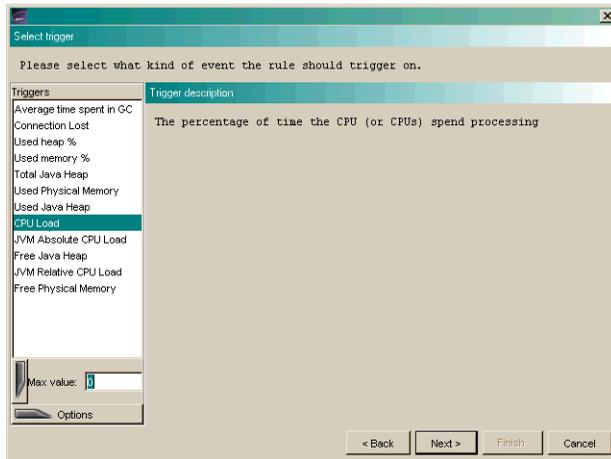
Figure 3-24 Name Rule Dialog Box



2. Enter the name of the new rule in Rule name: and click Next.

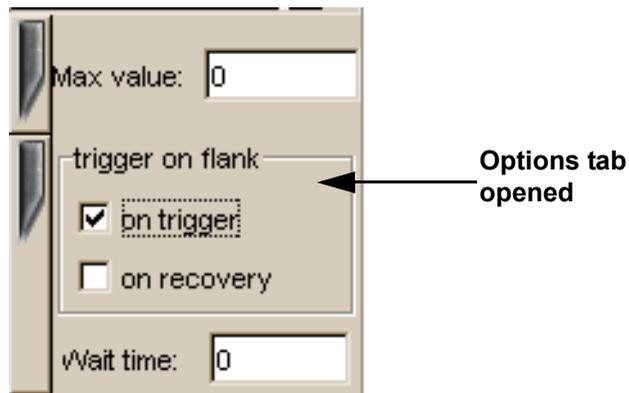
The Select trigger dialog box appears (Figure 3-25):

Figure 3-25 Select Trigger Dialog Box



3. Select a trigger (the individual triggers are described in the right panel).
4. Enter a threshold in the text box below the trigger list, if required (Figure 3-26; this box will be marked either Min value or Max value, depending on the type of trigger selected).

Figure 3-26 Trigger Threshold and Options Text Boxes

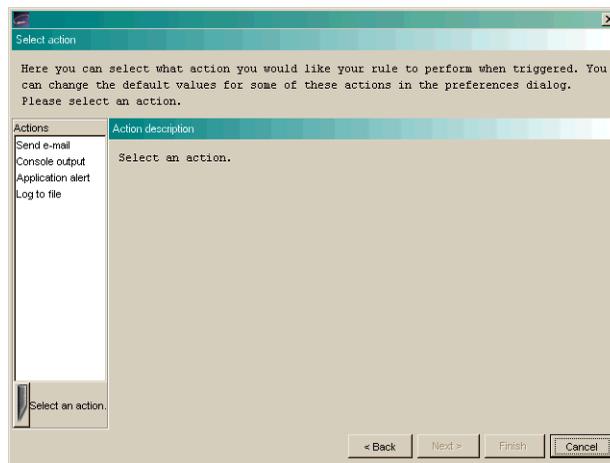


5. Select further options under the Option tab. For example, in Figure 3-26, you need to select what kind of aspect value change will trigger the notification:

- on trigger, which triggers the notification when the aspect reaches the trigger value from a lower value (for example, if the trigger is 80 and the aspect value moves up from 75).
 - on recovery, which triggers the notification when the aspect reaches the trigger value from a higher value (for example, if the trigger is 80 and the aspect value moves down from 85).
6. Click Next.

The Select Action dialog box appears (Figure 3-27):

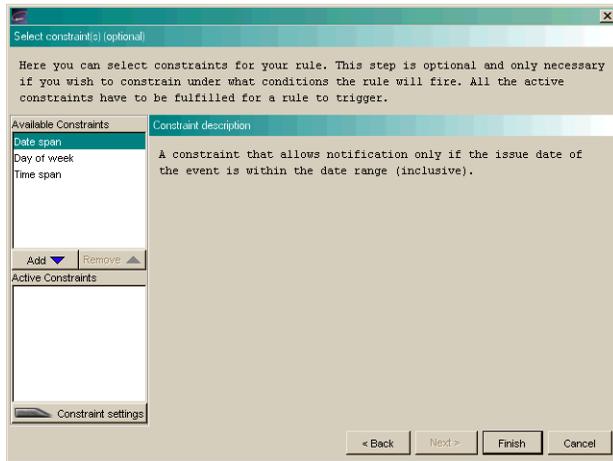
Figure 3-27 Select Action Dialog Box



7. Select an action and enter settings data, if required.
8. If necessary, add a constraint to the rule (this step is optional; if you don't want to add a constraint, go to step 8):
 - a. Click Next.

The Add Constraint dialog box appears (Figure 3-28):

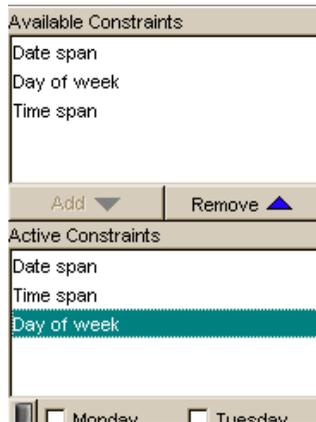
Figure 3-28 Add Constraint Dialog Box



- b. Select a constraint and click Add.

The constraint name will appear in the add list, as shown in [Figure 3-29](#).

Figure 3-29 Constraint Added



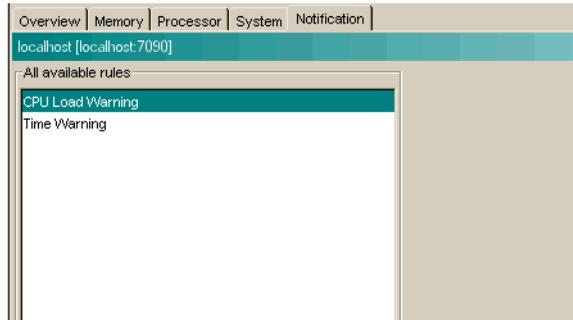
**Constraint settings;
Day of week
selected.**

- c. Enter constraint settings in the text fields under the list of constraints ([Figure 3-29](#)).

9. Click Finish.

The new rule appears in the All available rules list on the Notification tab, as shown in [Figure 3-30](#).

Figure 3-30 New Rule in List



10. Add the rule to your connection as described in [“Add a Rule to WebLogic JRockit 7.0 JVM”](#) on page 29

Editing a Rule

To edit a rule, do the following:

1. In the Available rules list, select the rule to be edited and click Edit Rule.
2. Check the name of the rule, edit it, if necessary, and click Next.
3. Check the trigger and trigger settings, edit them, if necessary, and click Next.
4. Check the action and the action settings and edit them if necessary.
5. To continue editing the rule, the do the following (optional; if you don't want to add a constraint, go to step 6):
 - a. Click Next.
 - b. Check the constraints and the constraint settings. Edit them, if necessary.
6. To finish the editing a rule, click Finish.

Add a Rule to WebLogic JRockit 7.0 JVM

To add a rule to WebLogic JRockit 7.0 JVM, do the following:

1. Select the rule to be added in the Available rules list.
2. Click Add to JRockit.

The rule appears in the Active rules for this connection list, as shown in [Figure 3-31](#).

Figure 3-31 Rule Added to Active rules for This Connection List



Remove a Rule from WebLogic JRockit 7.0 JVM

To remove a rule from WebLogic JRockit 7.0 JVM, do the following:

1. Select the rule to be removed in the Active rules for this connection list.
2. Click Remove from JRockit.

The rule will now be removed from the Active rules for this connection list.

Remove a Rule

To remove a rule from the Available rules list, do the following:

1. Select the rule to be removed.
2. Click Remove Rule.
A removal confirmation dialog box appears.
3. Click Yes
4. The rule disappears from the Available rules list.

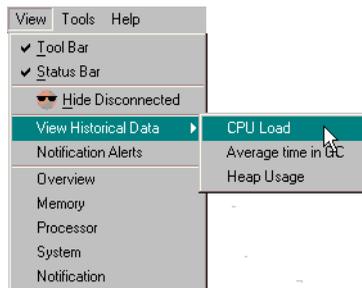
View Historical Data

The historical data window displays a chart where historical data for an aspect can be viewed. This is useful for observing trends over time and, for example, finding when a server running with WebLogic JRockit 7.0 JVM has its peak loads.

To open this window, do the following:

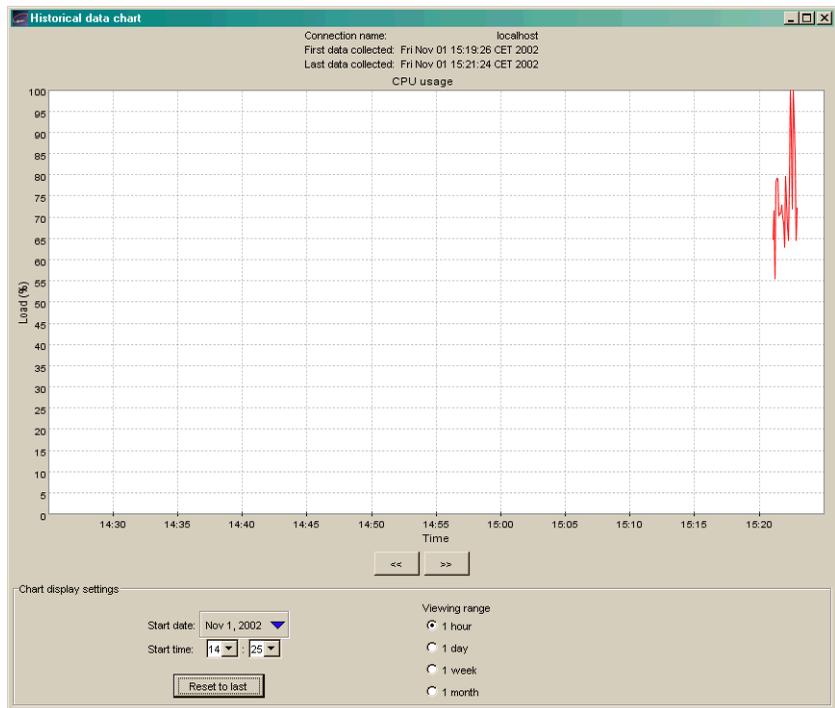
1. Select the connection for which you want to view data.
2. Open the View menu and select View Historical Data.
3. Select the aspect for which you want to view historical data, as shown in [Figure 3-32](#).

Figure 3-32 View Menu with Historical Data Submenu Open



Historical data for the selected aspect appears ([Figure 3-33](#)).

Figure 3-33 Historical Data (CPU Load Selected)



4. Navigate through time either by using the arrows or changing the start time in the Chart display settings.

To be able to observe historical data, aspect data from WebLogic JRockit 7.0 JVM must first have been persisted, that is, written to file. See [“Setting Other Preferences” on page 12](#) to enable or disable persistence. The following aspects are possible to persist, and thus display, historical data for:

- Used heap (as a percentage)
- CPU load (as a percentage)
- Average time spent garbage collecting (as a percentage)

As soon as data has been created by a connected connection, it is available for historical observation.

Using Advanced Features of the Console

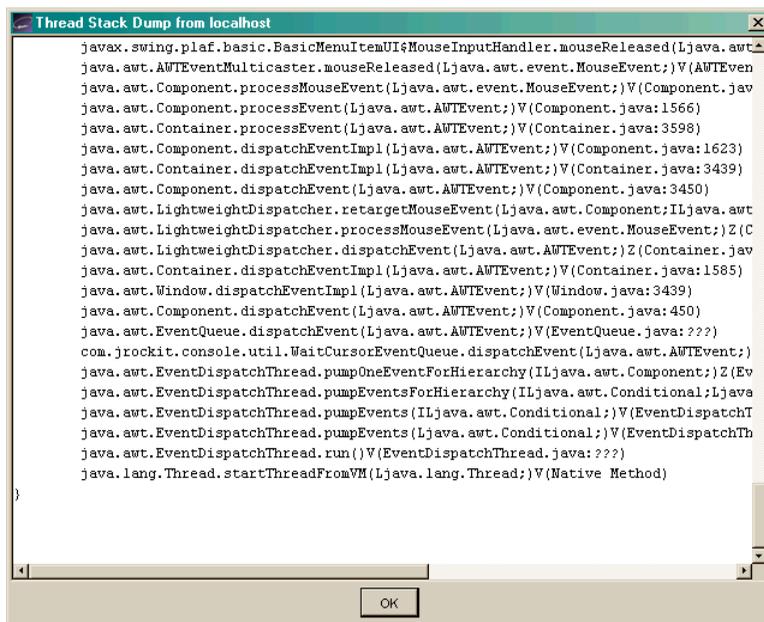
This section describes the more advanced features of the Management Console. Some of these are only available when running in the Developer mode, described in “[Setting the Operation Mode](#)” on page 11.

View Thread Stack Dump

The stack dump contains a list of all running threads in WebLogic JRockit 7.0 JVM with a method call stack trace for each thread.

To view the thread stack dump, open the Tool menu and select View Thread Stack Dump. A dialog box containing the stack dump appears ([Figure 3-34](#)).

Figure 3-34 Thread Stack Dump



```

Thread Stack Dump from localhost
javafx.swing.plaf.basic.BasicMenuItemUI$MouseInputHandler.mouseReleased(Ljava.awt.
java.awt.AWTEventMulticaster.mouseReleased(Ljava.awt.event.MouseEvent;)V(AWTEvent
java.awt.Component.processMouseEvent(Ljava.awt.event.MouseEvent;)V(Component.jav
java.awt.Component.processEvent(Ljava.awt.AWTEvent;)V(Component.java:1566)
java.awt.Container.processEvent(Ljava.awt.AWTEvent;)V(Container.java:3598)
java.awt.Component.dispatchEventImpl(Ljava.awt.AWTEvent;)V(Component.java:1623)
java.awt.Container.dispatchEventImpl(Ljava.awt.AWTEvent;)V(Container.java:3439)
java.awt.Component.dispatchEvent(Ljava.awt.AWTEvent;)V(Component.java:3450)
java.awt.LightweightDispatcher.retargetMouseEvent(Ljava.awt.Component;ILjava.awt
java.awt.LightweightDispatcher.processMouseEvent(Ljava.awt.event.MouseEvent;)Z(C
java.awt.LightweightDispatcher.dispatchEvent(Ljava.awt.AWTEvent;)Z(Container.jav
java.awt.Container.dispatchEventImpl(Ljava.awt.AWTEvent;)V(Container.java:1585)
java.awt.Window.dispatchEventImpl(Ljava.awt.AWTEvent;)V(Window.java:3439)
java.awt.Component.dispatchEvent(Ljava.awt.AWTEvent;)V(Component.java:450)
java.awt.EventQueue.dispatchEvent(Ljava.awt.AWTEvent;)V(EventQueue.java:???)
com.jrockit.console.util.WaitCursorEventQueue.dispatchEvent(Ljava.awt.AWTEvent;)
java.awt.EventDispatchThread.pumpOneEventForHierarchy(ILjava.awt.Component;)Z(Ev
java.awt.EventDispatchThread.pumpEventsForHierarchy(ILjava.awt.Conditional;Ljave
java.awt.EventDispatchThread.pumpEvents(ILjava.awt.Conditional;)V(EventDispatchT
java.awt.EventDispatchThread.pumpEvents(Ljava.awt.Conditional;)V(EventDispatchTh
java.awt.EventDispatchThread.run()V(EventDispatchThread.java:???)
java.lang.Thread.startThreadFromVM(Ljava.lang.Thread;)V(Native Method)
}
OK

```

Method Profiling Tab

Note: You must be in the developer operation mode before you can perform the tasks described in this section. For more information on entering the developer operation mode, see [“Setting the Operation Mode” on page 11](#)

The Method Profiler tab allows the developer to monitor method execution in a non-intrusive way. The Method Profiler can provide information about the average time spent in selected methods and the number of times methods are invoked.

Method Templates are collections of methods that can be re-used on different connections. There is a Default template, but the user may also create new templates.

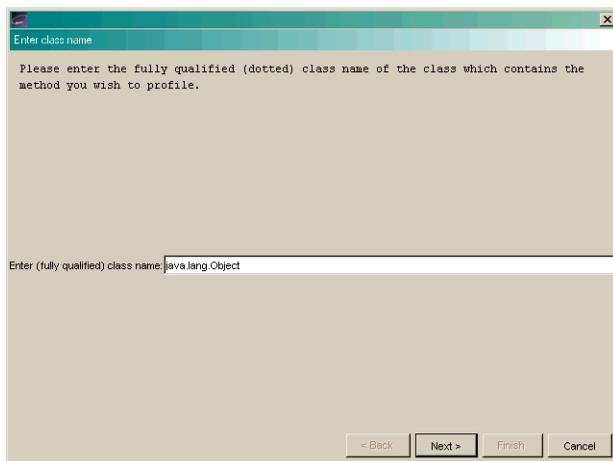
Adding a Method to a Template

To add a method to a template, do the following:

1. Select the template to be modified from the Select template list.
2. Click Add Method.

The Enter class name dialog box appears ([Figure 3-35](#)).

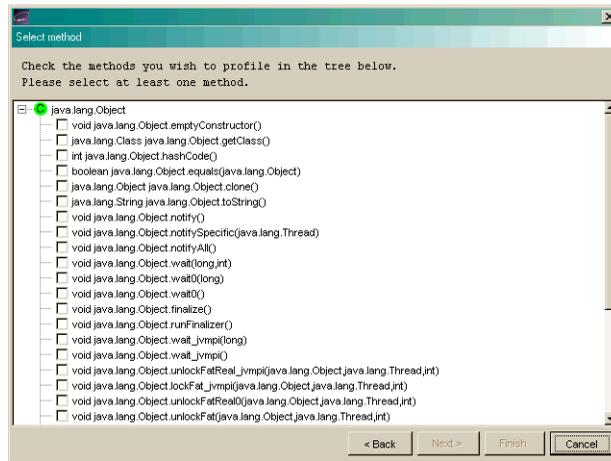
Figure 3-35 Enter Class Name Dialog Box



3. Enter a fully qualified class name, for example, `java.util.Vector`, in the text field and click Next.

The Select method dialog box appears (Figure 3-36):

Figure 3-36 Select Method Dialog Box



4. Select the methods to be added to the template and press Finish.

The method name will appear on the Method profiling information list, as shown in Figure 3-37.

Figure 3-37 Method Profiling Information List with Method Added

Method profiling information	
Method	Invocation count
StringBuffer StringBuffer.append(String)	
void Object.wait0(long)	
void System.arraycopy(Object,int, Object,int,j,...	

Removing a Method from a Template

To remove a method from a template, do the following:

1. From the Select template list, select the template you want to modify.
2. From the Method Profiling Information list, select the method(s) to be removed from the template.
3. Click Remove Method.

Creating a New Template

To create a new template, do the following:

1. Click New template.

The New template dialog box appears (Figure 3-38).

Figure 3-38 New Template Dialog Box



2. Enter a name for the new template in the text field.
3. Click OK.

Removing a Template

To remove a template, do the following:

1. From the Select template list select the template to be removed.
2. Click Remove.

A confirmation dialog box appears.

3. Click Yes.

Starting and Stopping Method Profiling

To start the method profiling, do the following:

1. From the Select template list, select the template to be started.
2. Click Start/Stop.

If you select Start, numbers in the Invocation count cells for each method begin to increment as method calls are made. If you select Stop, this activity will cease.

Method Profiling Settings

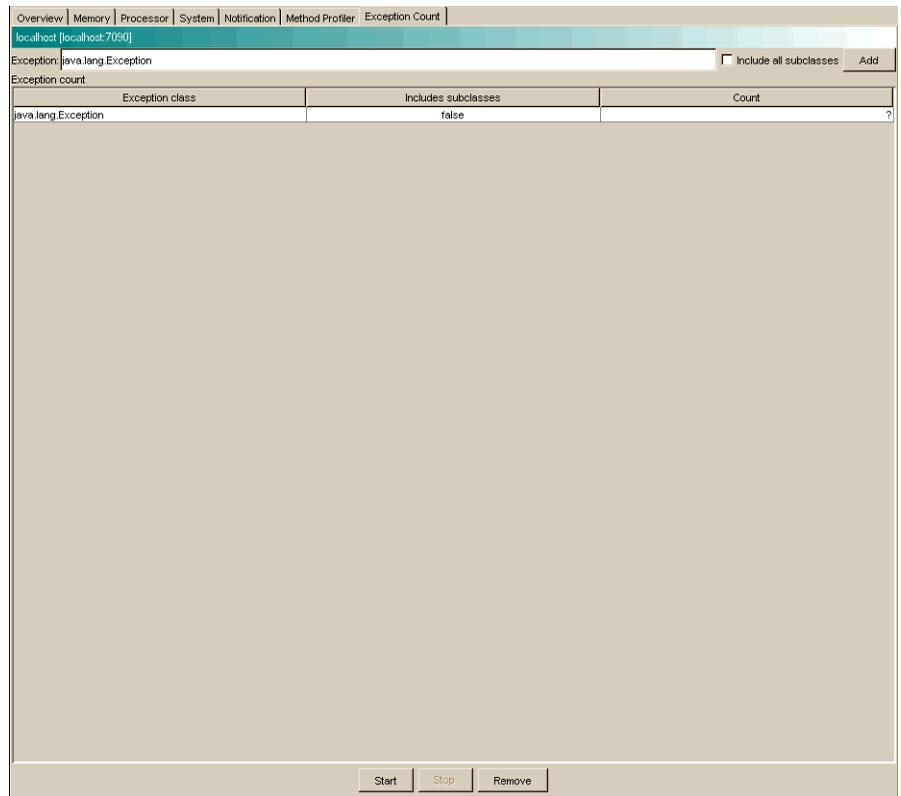
You can switch between using qualified method names or short names in the method profiling table.

- To enable invocation count, select the Invocation count checkbox at the bottom of the page.
- To enable timing, select the Timing checkbox at the bottom of the page.

Exception Counting Tab

The Exception Count tab ([Figure 3-39](#)) shows exceptions thrown in WebLogic JRockit 7.0 JVM. It counts the number of exceptions of a certain type thrown.

Figure 3-39 Exception Counting Tab



Add an Exception

To add an exception to observe, do the following:

1. Enter the fully qualified name of the exception into the text field at the top of the page, e.g., “`java.io.IOException`”.
2. Choose whether or not all subclasses of that exception should be included in the count by selecting or deselecting the Include subclasses checkbox.
3. Click Add. You can only add subclasses of `java.lang.Throwable` which are loaded in WebLogic JRockit 7.0 JVM and you can only add exceptions while connected.

The exception should now be displayed in the table.

Starting, Stopping, and Removing an Exception Count

To start the exception count, click Start. The results should now appear next to the name of the exception being counted. Similarly, to stop the exception count, click Stop.

To remove an exception from the count, select the exception to be removed and click Remove.

Closing the Console

To close the Management Console and disconnect all connections, open the Connection menu and select Exit. Clicking X in the top right corner of the window will also close the Management Console.

3 *Using the WebLogic JRockit 7.0 JVM Management Console*
