BEA JRockit® Mission Control™

Method Profiling

JRockit Mission Control 3.0.2
Document Revised: June, 2008

Contents

CHAPTER 1

Introduction to Profiling Methods and Using Exception Counters

The Management Console's profiling feature allows you to, non-intrusively, profile methods invoked and exceptions thrown during application runtime. You can learn much about an application through profiling; for example, you can see how often a method is invoked, how long it takes a method to execute during runtime, or how often specific exceptions are thrown. This sort of information is useful in determining whether or not you are getting optimal performance from your application. By studying the profile of an application run, you can make informed decisions about how you've configured and tuned the JVM running the application.

Profiling Tabs

Application profiling is done on two different Management Console tabs:

- Method Profiler Tab; see Using the Method Profiler.
- Exception Count Tab; see Using the Exception Counter.

CHAPTER 2

Using the Method Profiler

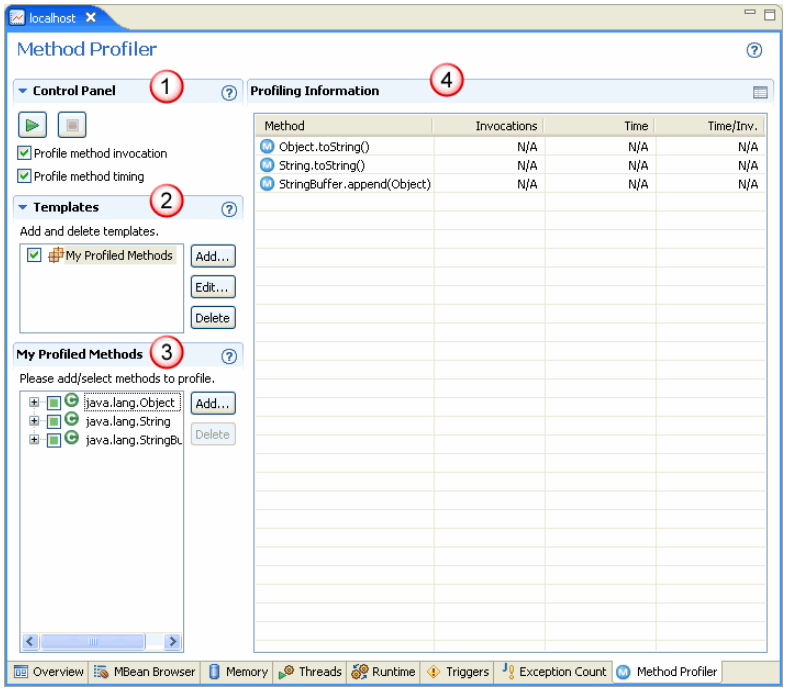
The Method Profiler tab allows you to monitor method execution in a non-intrusive way. Method profiling can provide information about the average time spent in selected methods and the number of times methods are invoked. monitor you running application's methods and find out where in the code you might have glitches.

The topics in this section include:

Getting Familiar with the Method Profiler Tab

Methods are profiled on the **Method Profiler** tab (Figure 2-1).

Figure 2-1 Method Profiler Tab



The **Method Profiler** tab is divided into the following sections:

1. **Control Panel**—from which profiling instructions are set and profiling is started and stopped.
2. **Templates**—where you can add and delete profiling template. Templates are collections of different methods you want to profile.
3. **My Profiled Methods**—lists all methods that are included in the template. This panel's name changes dynamically to correspond to the selected template in the **Templates** panel.
4. **Profiling Information**—which shows method-by-method results of the profile.

Profiling the Methods

This section describes how to run the Method Profiler. Before you begin, make sure you have completed these steps:

1. Created a template, as described in Working with Templates.

2. Added methods to the template, as described in Working with Methods.

To profile methods

1. In the **Control Panel**, select the profiling parameters you want to use (you must select at least one parameter).
   - **Profile method invocation**, which will display how many times the method has been invoked since the profiling started.
   - **Profile method timing**, which will display how much time (in milliseconds) it takes for each method to execute.
2. In the **Templates** panel, click the checkbox next to the template you want to use.

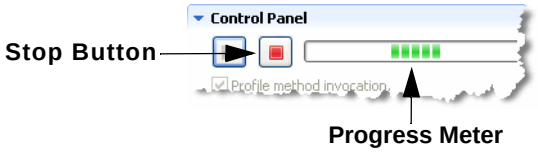
   That template name will appear atop the list of the methods.
3. Optionally, if you want to add new methods to the template, in the list of methods, open the class you want to profile and select the methods you want to add to the profile.
4. In the **Control Panel**, click the start button to begin profiling (Figure 2-2).

Figure 2-2 Method profiler start arrow

When profiling commences, a progress meter will appear in the Control Panel (Figure 2-3).

5. Click the stop button to stop the profiling (Figure 2-3).

Figure 2-3 Method profiler progress meter and stop button



The results of the profile will appear in the **Profiling Information** panel.

Working with Templates

Templates are a user-defined collection of methods you want profiled during runtime. By using templates, the same methods are profiled during a run, eliminating the need to reselect methods

each time you want to profile the same methods. This not only saves time, but it ensures consistency and accuracy of the profile results. This section shows how to use the **Method Profiler** tab by setting up your own templates.

The following topics are covered:

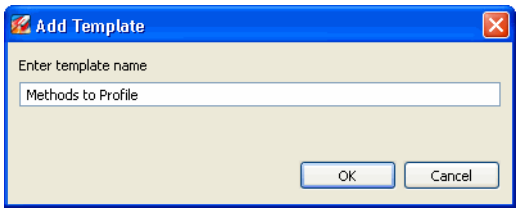
- To create a template
- To rename a template
- To delete a template

To create a template

1. In the **Templates** panel, click **Add**

   The **Add Template** dialog box appears (Figure 2-4).

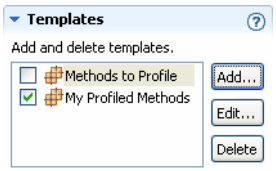
Figure 2-4 Add Template dialog box



2. Enter a name for your template (for example, `Methods to Profile`) and click **OK**.

   The new template will appear in the Templates list (Figure 2-5).

Figure 2-5 New template added



To add methods to the template, see To add methods to the template.

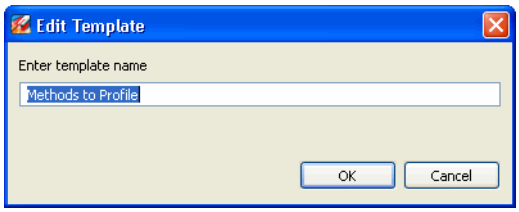
To rename a template

1. Highlight the template you want to rename.

2. Click **Edit**.

   The **Edit template** dialog box appears (Figure 2-6).

Figure 2-6Edit Template dialog box



3. Type in a new name for the template.
4. Click **OK**.

   The renamed template will appear on the **Templates** list.

To delete a template

1. In the **Templates** panel, select the template you want to delete.
2. Click **Delete**.

   The template disappears from the **Templates** panel.

Working with Methods

Templates are comprised of the methods you want counted. Use the list of methods panel:

- To add methods to the template
- To delete a method from the template

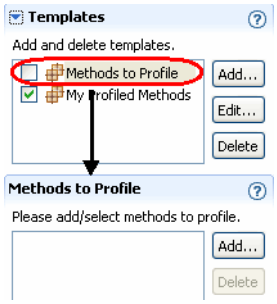
To add methods to the template

Before you can add methods, you need to create a template, as described in To create a template.

1. In the **Templates** panel, highlight the template to which you want to add methods.

   The title of the template list will change to match the name of the highlighted template (Figure 2-7).

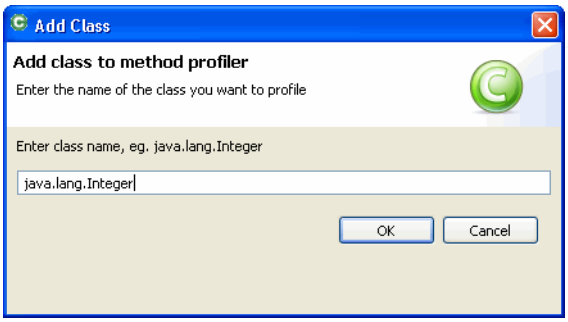
Figure 2-7 Method list panel with new template name



2. In the method list panel, click **Add**.

   The **Add Class** dialog box appears (Figure 2-8).

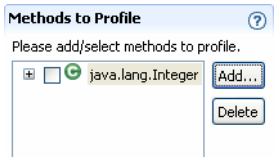
Figure 2-8 Add Class dialog box



3. Type the name of the class you want to profile (for example, `java.lang.Integer`) and click **OK**.

   The new class appears in the method list (Figure 2-9).

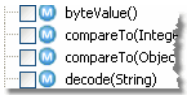
Figure 2-9 Method list with class added



4. Click the plus sign (+) next to the class name to see the methods that are included in that class.

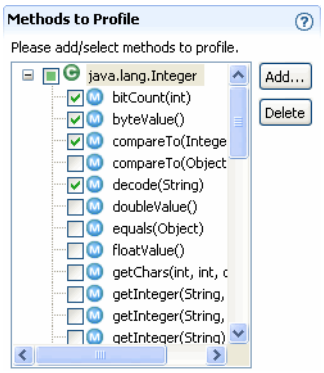
The list expands to show all methods in that class. The methods are indicated by an **M** graphic (Figure 2-10).

Figure 2-10 M graphics indicate method



5. Click the checkbox next to each method you want to profile (Figure 2-11).

Figure 2-11 List with methods selected



The selected methods will be profiled when method profiling is launched, as described in Profiling the Methods.

To delete a method from the template

1. In the list of methods, highlight the method you want to delete.
2. Click **Delete**.

   The method disappears from the list.

Jumping to Application Source

If you are using the Method Profiler as an Eclipse plug-in, you can jump from the **My Profiled Methods** tree or the **Profiling Information** table directly to the source code. A feature called *Jump-to-Source* allows you not only to see the name of a "problem" method displayed in the **My Profiled Methods** tree or **Profiling Information** table, but lets you jump from the displayed method name directly to that method's source, where you can evaluate the code to see what might

be causing the problem. This feature extremely is useful in helping you locate and debug coding errors that are creating runtime problems for your application.

To jump to the source code from the Method Profiler

1. In the **My Profiled Methods** tree or **Profiling Information** table and right-click the problem method to open a context menu.
2. Select **Open Method**.
3. The source code appears in a separate editor.

CHAPTER 3

Using the Exception Counter

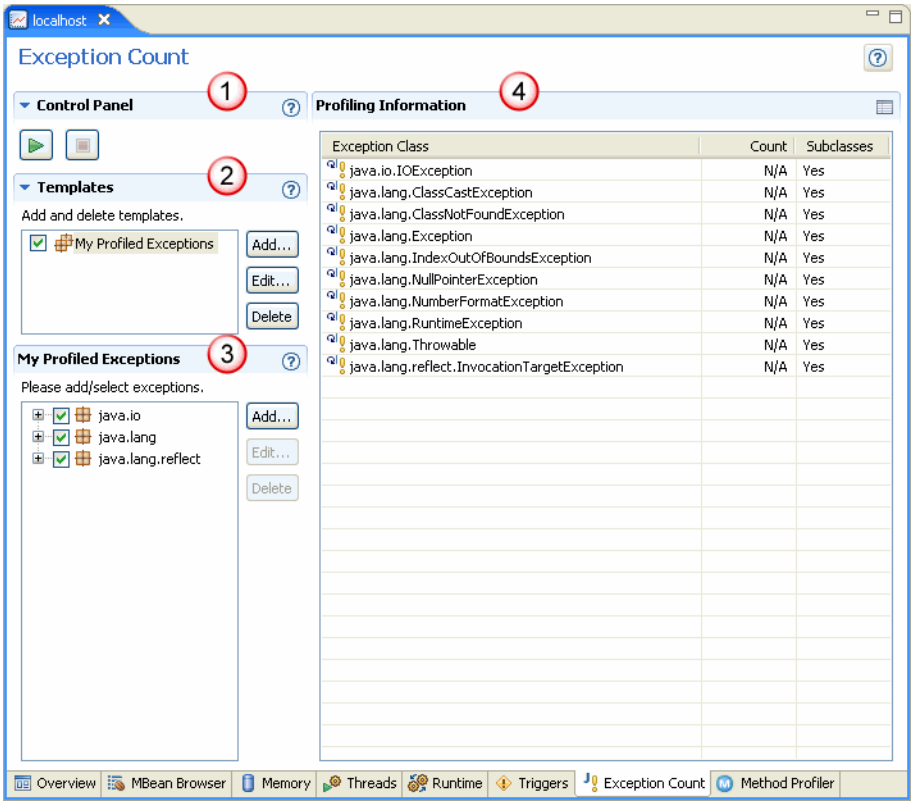
Exception counting is a type of profiling that enumerates exceptions thrown by the Oracle JRockit JVM during application runtime. As the name implies, exception counting counts the number of exceptions of a certain type, providing information that is helpful when you are troubleshooting your Java application.

This section includes the following topics:

Getting Familiar with the Exception Count Tab

Exceptions are counted on the **Exception Count** tab (Figure 3-1).

Figure 3-1 Exception Count Tab



The **Exception Count** tab is divided into the following sections:

1. **Control Panel**—where profiling is started and stopped.
2. **Templates**—where you can add and delete exception count templates. Templates are collections of different exception counters you want to profile.
3. **My Profiled Exceptions**—lists all exceptions that are included in the template. This panel name changes name dynamically to correspond to the selected template in the **Templates** panel.
4. **Profiling Information**—shows the exceptions thrown during the profiling session.

Profiling the Exceptions

This section describes how to run the Exception Counter. Before you begin, make sure you have completed these steps:

1. Created a template, as described in Working with Templates.
2. Added methods to the template, as described in Working with Exceptions in the Template.

To profile exceptions

1. In the **Templates** panel, click the checkbox next to the template you want to use.

   That template name will appear atop the list of exceptions.

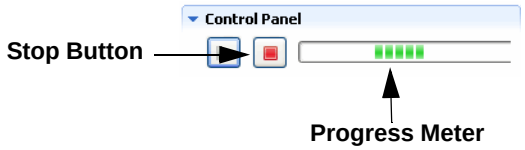
2. In the list of exceptions, open the exception class you want to profile and select the exceptions you want to count.
3. In the **Control Panel**, click the start arrow to begin profiling (Figure 3-2).

Figure 3-2 Exception counter start arrow

   When profiling commences, a progress meter will appear in the Control Panel (Figure 3-3).

4. When you have finished profiling, click the stop button (Figure 3-3).

Figure 3-3 Exception counter progress meter and stop button



   The results of the count will appear in the **Profiling Information** panel.

Jumping to Application Source

If you are using the Exception Counter as an Eclipse plug-in, you can jump from the **My Profiled Exceptions** table or the **Profiling Information** table directly to the source code. A feature called *Jump-to-Source* allows you not only to see the name of a "problem" class displayed in the **My Profiled Exceptions** table or **Profiling Information** table, but lets you jump from the displayed

class name directly to that class's source, where you can evaluate the code to see what might be causing the problem. This feature extremely is useful in helping you locate and debug coding errors that are creating runtime problems for your application.

To jump to the source code from the Exception Counter

1. In the **My Profiled Exceptions** table or the **Profiling Information** table, right-click the problem class to open a context menu.
2. Select **Open Type**.
3. The source code appears in a separate editor.

Working with Templates

Templates are a user-defined collection of exceptions you want counted during runtime. By using templates, the same exceptions are counted during a run, ensuring consistency and accuracy of the count results.

The following topics show how to use the templates on the **Exception Count** tab:

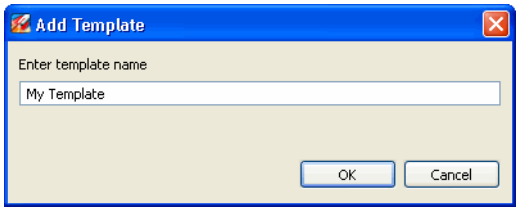
- To create a template
- To rename a template
- To delete a template

To create a template

1. In the **Templates** panel, click **Add**.

   The **Add Template** dialog box appears (Figure 3-4).

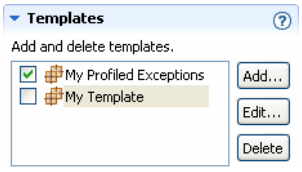
Figure 3-4 Add Template dialog box



2. Enter a name for your template (for example, `My Template`) and click **OK**.

   The new template will appear in the **Templates** list (Figure 3-5).

Figure 3-5 New Template added



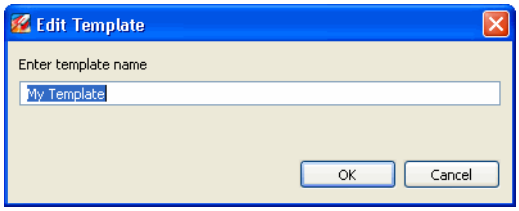
Add exceptions to the template, as described in Working with Exceptions in the Template.

To rename a template

1. Highlight the template you want to rename.
2. Click **Edit**.

   The **Edit Template** dialog box appears (Figure 3-6).

Figure 3-6 Edit Template dialog box



3. Click **OK**.

   The renamed template will appear on the Templates list.

To delete a template

1. Highlight the template you want to delete.
2. Click **Delete**.

   The template name disappears from the list of templates.

Working with Exceptions in the Template

Templates are comprised of the exceptions you want counted. Use the list of exceptions panel.

- To add exceptions to the template

- To edit exceptions
- To remove exceptions from the template

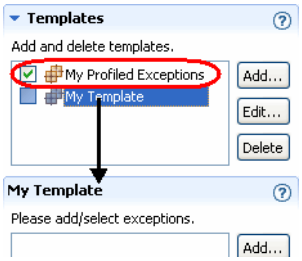
To add exceptions to the template

1. In the **Template** panel, highlight the template to which you want to add exceptions.

   The title of the template list will change to match the name of the highlighted template (Figure 3-7).

   Note: If the template name is too long, the list name will be truncated.

Figure 3-7 Exception list panel with new template



2. In the exception list panel, click **Add**.

   The **Add new exception to profile** dialog box appears (Figure 3-8).

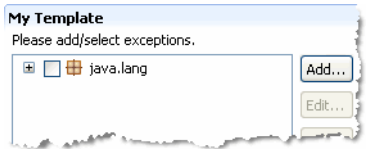
Figure 3-8 Add new exception to profile

3. Type the name of the exception class you want to use (for example, `java.lang.Exception`).
4. If you want to include all subclasses to the specified exception class in the exception count profile, select **Include subclasses in the exception count**.

5. Click **OK**.

   The dialog box closes and the new exception class appears in the exception list (Figure 3-9).

Figure 3-9 Exception list



6. Click the plus sign (+) next to the class name.

   The list will expand and you will see all subclasses for the selected class

7. Click the checkbox next to each subclass you want to profile.

   The selected exceptions will be profiled when exception profiling is launched, as described in Profiling the Exceptions.

To edit exceptions

Editing an exception allows you to enabled or disable profiling of subclasses. The checkbox in the tree enables or disables the actual profiling but in the dialog box, you can specify whether or not classes derived from that exception class should be included in the count or not.

1. Highlight the exception you want to edit.
2. Click **Edit**.

   The Edit <*exception.name*> dialog box appears (Figure 3-10).

Figure 3-10Edit exception dialog box

3. Do one of the following:
   - To include classes derived from the exception, select the checkbox.
   - If classes are already included and you want to exclude them, deselect the checkbox.
4. Click **OK**.

To remove exceptions from the template

1. Select the exception class or subclass you want to delete.
2. Click **Delete**.

   The exception is removed from the list.