

---

## NAME

mrprof - (Method R Profiler) profile the duration of an Oracle application program execution

## SYNOPSIS

```
mrprof [ --advisory ] [ --cdefs=file ] [ --debug=number ] [ --encoding=string ] [ --error-emit=number ]
[ --etc=dir ] [ --etcuri ] [ --first-member ] [ --fix-tim-values ] [ --force-sql-formatting ] [ --help or -? ]
[ --html ] [ --ignore-multiple-sessions ] [ --ignore-multiple-sessions-file=file ] [ --indent ] [
--jparam=string ] [ --license ] [ --man ] [ --max-java-heap=size ] [ --member=file ] [ --noelide ] [
--ofile=string ] [ --open ] [ --param=string ] [ --params-file=file ] [ --patch-negative-counts ] [
--progress=file ] [ --script=string ] [ --script-comment=string ] [ --script-extension=string ] [
--shareable ] [ --sparam=string ] [ --sql-text-width=number ] [ --stash=number ] [ --trace=string ] [
--trcsex ] [ --trcunit=number ] [ --usage ] [ --verbose=number ] [ --version ] [ --xsd=file ] file...
```

## DESCRIPTION

**mrprof** (Method R Profiler) is a fast, high-precision response time profiler for Oracle Database application programs. It reads Oracle Database event 10046 extended SQL trace data to produce an HTML5 report that application developers, database administrators, and performance analysts use to learn how their program durations are being consumed. Profile reports are useful for code reviews, performance tests, concept proofs, hardware and software evaluations, upgrades, troubleshooting, ...any time it is important to understand where an Oracle application program's time is being spent.

**mrprof** accounts for every microsecond of time within its input to provide a complete explanation of where a program's time has gone. Other tools to which you may be accustomed, like Oracle's **tkprof**, do not. New **mrprof** users are often surprised by the ramifications of this sensitivity. When using **mrprof**, it is crucial to remember: **THE DURATION YOUR TRACE FILE EXPLAINS MUST MATCH THE DURATION OF THE END-USER EXPERIENCE YOU WANT TO DIAGNOSE.**

**mrprof** determines its action based upon the type of the file that is passed to it. If you specify a *trc* file, **mrprof** will produce an XML file and, by default, an HTML file. If you specify an *xml* file, **mrprof** will produce an HTML file. All other input file types (*gzip*, *zip*, *bz2*, *tar*) are assumed to be deflated representations of Oracle trace data, for which **mrprof** will produce an XML file and, by default, an HTML file. See **mrnunzip** for more information.

**mrprof** uses the same case-tolerance rules as its underlying operating system. For example, on a Linux system, *x.trc* and *X.trc* are different files, whereas on a Microsoft Windows system, the two file names are synonymous.

## OPTIONS

### --advisory

Print advisory text in the HTML profile report. The default is **--advisory**. Use **--noadvisory** to suppress the advisory text.

### --cdefs=file

Read call definitions from *file*. The default is *\$HOME/.method-r/workbench/8.0.5.0/cdefs.xml*.

### --debug=number

Write rudimentary diagnostic information to STDERR. Use this option only with the assistance of Method R product support. Using **--debug** is equivalent to using **--debug=1**.

### --encoding=string

Declare the encoding for the input trace file. The default is **UTF-8**. Choose from one of the following values:

UTF-8	UTF-16	
ISO-8859-1	ISO-8859-2	ISO-8859-3
ISO-8859-4	ISO-8859-5	ISO-8859-6
ISO-8859-7	ISO-8859-8	ISO-8859-9
ISO-2022-JP	Shift_JIS	EUC-JP

---

**--error-emit=*number***

Emit details for only *number* instances of a given error type into the HTML report. The default is 5. Use **--error-emit=0** to show all errors in detail. Using 0 can cause **mrprof** to consume a lot of memory and cause browser performance problems when viewing the HTML report.

**--etc=*dir***

Use *dir* for external references made to */etc* from the output HTML. The default value is `$MRWORKBENCH_INSTALL/etc`.

**--etcuri**

Canonicalize the **--etc** value to a proper URI. Use **--noetcuri** to not canonicalize the value. The default is **--etcuri**.

**--first-member**

Process only the first non-empty regular file of a container (for example, a zip file). The default is **--first-member**. Using **--nofirst-member** creates a profile for the concatenation of all the members of the container.

**--fix-tim-values**

Attempt to fix out-of-order tim values. The default is **--nofix-tim-values**.

**--force-sql-formatting**

Format the text of a SQL statement even if it appears to already be formatted. The default is **--noforce-sql-formatting**.

**--help** or **-?**

Print usage synopsis and options information, and exit.

**--html**

Create the HTML report, in a *.html* file with the same base name as the input file. The default value is **--html**. Use **--nohtml** if you want not to create HTML output.

**--ignore-multiple-sessions**

Ignore trace data for sessions whose session ID and serial number are not identical to the first session ID and serial number found in the trace file. The default is **--ignore-multiple-sessions**. To process multi-session data (we do not recommend doing that), use **--noignore-multiple-sessions**.

**--ignore-multiple-sessions-file=*file***

Write input trace lines that **mrprof** doesn't recognize, and that **mrprof** is ignoring (see **--ignore-multiple-sessions**) to *file*. The default is not to write ignored lines.

**--indent**

Indent HTML output. The default is **--noindent**. Indenting HTML is useful primarily for HTML debugging purposes. Note specifically that using **--indent** will cause **mrprof** to generate much larger HTML files and prohibit **mrprof**'s SQL formatter from doing its job properly.

**--jparam=*string***

Send the option specified in *string* to the Java Virtual Machine. You may use this option multiple times on a single command line. Use **--jparam=-?** to see information about which parameters are allowed.

**--license**

Display license content and exit.

**--man**

Print the manual page and exit.

**--max-java-heap=size**

Shorthand for **--jparam -Xmxsize**, where *size* is expressed in bytes. *size* must be a multiple of 1024 bytes. Append k or K to indicate kibibytes (multiples of 1024), or m or M to indicate mibibytes (multiples of 1048576). For example, **--max-java-heap=1900m** means to set the max Java heap size to 1,900 mibibytes. The default value is **1000m**. See <http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/java.html> for details on how large your Java heap can be for your platform.

**--member=file**

Name of the member (individual file) to read from the container (for example, a zip file). If you specify a member name with the option, **mrprof** will ignore the **--first-member** option. The default is no filename.

**--noelide**

Shorthand for **--param=PROFILE-DISPLAYED-THRESHOLD=0**. Use this option to show all the details of every subroutine call and every SQL and PL/SQL statement in the profile report.

**--ofile=string**

Use *string* as a format string for the output file base name. Where you specify the string "%s", **mrprof** will insert the base name of the input file. If you do not specify "%s" in *string*, then **mrprof** will append it to the end of the *string* you specify. The default is **--ofile=%s**.

This flexible mechanism allows you to choose output filename prefixes or suffixes. A prefix can include a path name. For example, if your command is **mrprof ora\_1492.trc**, the relationship between **--ofile** value and output file name is shown in the following table:

<b>--ofile</b> value	output file name
-----	-----
%s-spec	ora_1492-spec.html
1-%s-spec	1-ora_1492-spec.html
abc-	abc-ora_1492.html
../	../ora_1492.html
../output/%s	../output/ora_1492.html
d:\tmp\mrprof\	d:\tmp\mrprof\ora_1492.html

**--open**

Open each output HTML file after creating it. The default value is **--open** if you provide only one input file name, or **--noopen** if you provide multiple input file names.

**--param=string**

Use *string* to override a *prof.xml* parameter setting. The string must have the form *name=value*; for example, **--param=PROFILE-DISPLAYED-THRESHOLD=0.01**. You may specify multiple name-value pairs by using separate **--param** option specifications on a single command line.

**--params-file=file**

Read transformation parameters from *file*. The default value is `$HOME/.method-r/workbench/8.0.5.0/prof.xml`.

**--patch-negative-counts**

Manipulate parent call {p,cr,cu} values that are less than or equal to the sum of the call's children's corresponding {p,cr,cu} values. The default value is true. Use **--nopatch-negative-counts** to not manipulate values.

**--progress=file**

Update *file* with a percentage representing how much input trace data has been converted into XML form.

**--script=*string***

Use either Microsoft Windows or Unix style scripting. Valid *string* values are **mswin** and **unix** (values are not case sensitive). This option sets the individual **--script-\*** options as follows:

<i>string</i>	<b>--script-extension</b>	<b>--script-comment</b>
mswin	".bat"	"rem"
unix	" "	"#"

Note that the effects of **--script** can be overridden with the individual **--script-\*** options. The default value is **MSWin**.

**--script-comment=*string***

Use *string* as the comment designator for any scripts this program will generate (for example, by using the **--trcsess** option). The default value is **rem**.

**--script-extension=*string***

Use *string* as the filename extension for any scripts this program will generate (for example, by using the **--trcsess** option). The default value is **.bat**.

**--shareable**

Create HTML output with embedded graphics and CSS that will render properly on someone else's system, even if that person doesn't have the */etc* directory contents that you do. The default value is **--shareable**. Use **--noshareable** to render HTML with links instead of embedded graphics and CSS. Using **--noshareable** will make HTML files that are smaller but less convenient.

**--sparam=*string***

Send the option specified in *string* to the Saxon XSLT processor. You may use this option multiple times on a single command line. Use **--sparam=?** to see information about which parameters are allowed.

**--sql-text-width=*number***

Format SQL statement text so that each line contains **number** or fewer bytes. The default is 180.

**--stash=*number***

Distinguish WAIT lines for a given cursor between no more than this **number** of intervening dbcalls so that if some of this cursor's WAIT lines belong to some missing dbcall, the WAIT lines will be properly attributed. The default is 100.

**--trace=*string***

Write diagnostic information about the input file processing to STDERR. Use this option only with the assistance of Method R product support. The type of information written is controlled by zero or more of the following keys (characters) in *string*:

Key	Writes...
*	all tracing information
l	basic diagnostic information
c	character set conversion setup info
d	check tim drift
h	shared statement identification details
l	input lines
s	synthetic dbcall details
t	input lines with t0 values appended
u	unaccounted-for time details
U	unaccounted-for time summary

---

w WAIT line details

For example, **--trace=ul** will show input lines annotated with unaccounted-for details.

#### **--trcsess**

Generate a script that will run Oracle **trcsess** commands to split a trace file containing information about two or more Oracle sessions into a set of files containing information about exactly one Oracle session per trace file. The script file base name is *f-trcsess*, where *f* is the base name of the input data being processed. The script file extension is determined by the setting of **--script-extension**. The default value is **--trcsess**. No suppress the **trcsess** script generation, use **--notrcsess**.

#### **--trcunit=number**

Regard time and duration values in the raw trace file as being expressed in units of *number* seconds. The default value is **.000001**. Oracle Database versions since 9.0 express all trace file time and duration values in microseconds, for which you should use **--trcunit=.000001**, the default. However, on some platforms prior to version 11.2, the Oracle kernel reports time and durations in 1,024-nanosecond units. On these platforms, you should use **--trcunit=.000001024**.

#### **--usage**

Print usage synopsis and exit.

#### **--verbose=number**

Print additional diagnostic information about **mrprof** on STDERR. Specifying **--verbose** with no numerical argument is equivalent to specifying **--verbose=1**. The default value is **--verbose=0**, which suppresses diagnostic information.

#### **--version**

Print the version number and exit.

#### **--xsd=file**

Emit *file* as an XML schema definition reference into the output XML file if *file* is a non-empty string. The default value is the empty string.

## TRANSFORM PARAMETERS

**mrprof** gives you access to several transform parameters that allow you to govern the appearance of your profile reports. You can specify these values either on the command line with the **--param** option, or you can record the values you want permanently in the *prof.xml* file specified by the **--params-file** option. The syntax of this file is:

```
<params>
  <param name="LOCALE-CODE">D</param>
  <!--
  XML comment syntax lets you disable settings without deleting them.
  <param name="CONTRIBUTIONS-MINIMUM-LIST-LENGTH">0</param>
  -->
</params>
```

If a parameter is not listed in the *prof.xml* file, **mrprof** will use its default value. **mrprof** recognizes the following parameters:

#### **ADVISORY**

Print advisory paragraphs that narrate the data in the profile report. The default value is **true**. To suppress the advisory paragraphs, use **false**.

#### **CONTRIBUTIONS-MINIMUM-LIST-LENGTH**

**mrprof** will render contributions in profiles that have at least this many rows in them, provided that this many contributing elements exist. To see all contributors for a given profile, set the value to **0**. However, be aware that rendering all contributions may seriously degrade transform runtime and memory performance. The default value is **4**.

**CREATE-SHAREABLE-HTML**

Create shareable HTML files by including images and styles directly within the HTML itself. The default value is **true**.

**DISTINCT-TEXT-THRESHOLD**

If the number of similar but distinct texts exceeds this value for a given statement, then **mrprof** highlights the distinct text count for the statement. The default value is **1**.

**EXECS-PER-PARSE-THRESHOLD**

If the application executes fewer than this quantity times the parse call count, then **mrprof** highlights the parse call count. The default value is **2**.

**LIBRARY-CACHE-MISSES-PER-STATEMENT-THRESHOLD**

If the number of Oracle database library cache misses is greater than this value for a given statement, then **mrprof** highlights the library cache miss count. The default value is **1**.

**LIOS-PER-ROW-THRESHOLD**

If the number of LIOs is greater than this value times the number of rows processed, then **mrprof** highlights the LIO count. The default value is **10**.

**LOCALE-CODE**

**mrprof** uses this value to determine its output number format. The domain of values and an example of each follows:

LOCALE-CODE	Format
-----	-----
A	123,456,789.000000
B	123 456 789.000000
C	123.456.789,000000
D	123 456 789,000000
E	123,456,789*000000 * decimal mark is &#0183; (a middot)
F	123 '456 '789,000000
G	12,34,56,789.000000
H	1,2345,6789.000000

The default value is **A**.

**PIOS-PER-LIO-THRESHOLD**

If the number of PIOs is greater than this value times the number of LIOs, then **mrprof** highlights the PIO count. The default value is **.1**.

**PROFILE-DISPLAYED-THRESHOLD**

**mrprof** will render details for any cursor or statement whose magnitude of contribution to total experience duration is greater than or equal to this value times the magnitude of the total experience duration. That is, if a contribution's elapsed time is  $e$ , the value of PROFILE-DISPLAYED-THRESHOLD is  $t$ , and the total experience duration is  $R$ , then **mrprof** will render details for the contribution only if  $\text{abs}(e) \geq t * \text{abs}(R)$ . It elides details for cursors and statements whose contributions fall short of this threshold. To see all contributors for a given profile, set the value to **-1**. However, be aware that rendering all contributions may seriously degrade transform runtime and memory performance. The default value is **.05**.

**PROFILE-RELEVANCE-THRESHOLD**

**mrprof** will highlight rows for elements whose magnitude of contribution to total experience duration is greater than or equal to this value times the magnitude of the total experience duration. That is, if a contribution's elapsed time is  $e$ , the value of PROFILE-RELEVANCE-THRESHOLD is  $t$ , and the total experience duration is  $R$ , then **mrprof** will highlight the contribution only if  $\text{abs}(e) \geq t * \text{abs}(R)$ . The default value is **.20**.

#### **ROWS-PER-EXEC-THRESHOLD**

If there are no Oracle fetch calls for a statement and the application manipulates fewer than this many rows per Oracle exec call, then **mrprof** highlights the exec call count. The default value is **2**.

#### **ROWS-PER-FETCH-THRESHOLD**

If the application retrieves fewer than this many rows per Oracle fetch call, then **mrprof** highlights the fetch call count. The default value is **2**.

#### **SHOW-CONFIDENTIALITY-LABEL**

If this value is **true**, then **mrprof** will render a confidentiality label at the top of the report. The default value is **false**.

#### **SHOW-CUMULATIVE-DURATIONS**

If this value is **true**, then **mrprof** will render columns showing cumulative duration information. Using **true** makes it easier to understand the relevance of a given row of data. Using **false** conserves horizontal space in the report, making it easier to print. The default value is **false**.

#### **SHOW-LITERAL-HIGHLIGHTING**

If this value is **true**, then **mrprof** will highlight literal values in SQL statements. The default value is **true**.

#### **SHOW-PROFILER-STATEMENT-ID**

If this value is **true**, then **mrprof** will render its internal statement id for each statement displayed in the report. The default value is **false**.

#### **SHOW-ZERO-VALUE-STRING**

If this value is **true**, then **mrprof** will render an unintrusive middot character into the report instead of actual zero values. The default value is **true**, which removes significant visual clutter from the rendered data.

#### **SQL-TEXT-PREFIX-LENGTH**

**mrprof** will render up to this many characters of statement text in each Profile by Cursor row. Decreasing the size of this number conserves more horizontal space in the report. The default value is **48**.

### **CALL SUBTYPES AND TOLERANCES**

**mrprof** uses highlighting to attract your attention to items in the report that you shouldn't overlook. For example if the average disk read latency for an experience is too high, **mrprof** will highlight it for you. But how does **mrprof** know how to define "too high"? That's where subtypes and tolerances come in.

**mrprof** highlighting rules are driven by configuration parameters that you can adjust. For example, on an old system you might want to highlight disk read calls that take longer than 10 ms (0.010 seconds). On a newer system, you might want to highlight disk read calls that take longer than 2 ms.

But in reality, things are even more complicated than that. There's no such thing as a single correct tolerance for disk read calls of all sizes. For example, imagine that your tolerance for a 2-block read is 4 ms. If a 2-block read takes 5 ms, you want to see a highlight on that tolerance violation. But if a 32-block read takes 5 ms, you probably don't want alarm bells going off. **mrprof** allows you to specify tolerances using a powerful sub-typing mechanism that can both highlight exceptional calls and allow you to group by call attributes.

**mrprof** subtype definitions are recorded in the *cdefs.xml* file specified by your **--cdefs** option. Here is an example of a **mrprof** call subtype definition:

```
<call name="db file scattered read">
  <subtype
    qualifier="$p3 &lt;= 16"
    qualifier-label="blocks &le; 16"
    tolerance="0.004"
  />
  <subtype
    qualifier="$p3 &lt;= 32"
    qualifier-label="16 &lt; blocks &le; 32"
    tolerance="0.008"
  />
  <subtype
    qualifier-label="blocks &gt; 32"
    tolerance="0.016"
  />
</call>
```

With the example definition shown above, whenever **mrprof** finds a call named "db file scattered read", it will evaluate the qualifier expressions (in document order) until it finds a match.

Qualifier expressions use Perl expression syntax (<http://perldoc.perl.org/perlop.html>). Because of XML syntax restrictions, you are not allowed to refer to three special operators in the normal way:

Operator	XML form
<	&lt;
>	&gt;
&	&amp;

You can refer to the following variables:

### **\$e**

The elapsed duration of the subroutine call.

### **\$p1, \$p2, \$p3**

The value of PARAMETER1, PARAMETER2, or PARAMETER3, respectively, for the given call. See your Oracle Database V\$EVENT\_NAME documentation for details.

Thus, "\$p3 &lt;= 16" in the example above means "the PARAMETER3 value is less than or equal to 16". For "db file scattered read" calls, PARAMETER3 is the number of blocks retrieved by an O/S read call, so the subtype definitions above will sub-type "db file scattered read" calls based on how many blocks they return:

- If the call returns 16 or fewer blocks, then **mrprof** will name the call "db file scattered read [blocks <= 16]" and use a latency tolerance of 0.004 seconds.
- Otherwise, if the call returns 32 or fewer blocks, then **mrprof** will name the call "db file scattered read [16 < blocks <= 32]" and use a latency tolerance of 0.008 seconds.
- Otherwise, **mrprof** will name the call "db file scattered read [blocks > 32]" and use a latency tolerance of 0.016 seconds.

In the call subtype definition syntax, each qualifier-label attribute is optional. If you specify a qualifier label, **mrprof** will append that label in [square brackets] to the end of the call name.

Likewise, each tolerance attribute value is optional. If you specify a tolerance, **mrprof** will use it to

determine whether to highlight the sub-typed call's mean call duration. If the mean duration of a sub-typed call exceeds the stated tolerance, then **mrprof** will highlight the mean duration value.

Note: **mrprof** does not honor the use of qualifier or qualifier-label attributes for synthetic calls like **CPU: PARSE dbcalls** and **unaccounted-for within dbcalls**. Use these attributes only with the Oracle timed event call names described in V\$EVENT\_NAME.

## SYNTHETIC CALLS

For "Profile by Subroutine" sections, **mrprof** may synthesize the following timed events:

### unaccounted-for between calls

Each time there is a duration between two temporally adjacent database or system calls that isn't exactly accounted for by the durations of some sequence of between-call Oracle timed events, **mrprof** tallies an "unaccounted-for between calls" event. Its duration is the amount required to account for the interval between the database or system calls.

### unaccounted-for within dbcalls

Each time an Oracle database call's measured elapsed time isn't exactly accounted for by the sum of the call's CPU time plus its time spent waiting for operating system calls, **mrprof** tallies an "unaccounted-for within dbcalls" event. Its duration is the amount required to account for the database call's exact duration.

### uncategorizable no calls

When **mrprof** finds more than one timestamp in a trace file but no dbcalls or syscalls anywhere within the file, it will report the difference in the outermost timestamps as "uncategorizable no calls".

### uncategorizable within ERROR dbcalls

### uncategorizable within PARSE ERROR dbcalls

ERROR and PARSE ERROR trace lines do not contain duration information. **mrprof** determines which lines preceding an error represent work done within the dbcall that raised the error. Although **mrprof** can unambiguously determine that time near the error has passed, it cannot determine the call's proportion of CPU duration, "unaccounted-for within dbcalls" duration, or "unaccounted-for between dbcalls" duration. For each such ERROR, **mrprof** will synthesize an "uncategorizable within ERROR dbcalls" event, and for each such PARSE ERROR, **mrprof** will synthesize an "uncategorizable within PARSE ERROR dbcalls" event

### uncategorizable within file

**mrprof** synthesizes one "uncategorizable within file" event for any trace file that contains a call whose begin time is earlier than the end time of a call that precedes it. Causes include:

- The trace file has been appended after an instance restart, and the tim values late in the file are smaller than the tim values early in the file.
- VKTM inaccuracies cause timings to appear out of order.
- Oracle Database Bug 752202 (fixed in 11.2) causes tim values to "jump back in time".

### uncategorizable within synthetic dbcalls

Sometimes, an Oracle Database kernel process writes trace data in which a sequence of one or more  $\text{dep} = n + 1$  database calls or system calls will not have a following  $\text{dep} = n$  parent. This can occur (i) at the end of a trace file, when tracing is deactivated during the execution of a  $\text{dep} = n + 1$  call for some  $n \geq 0$ ; or (ii) when **mrprof** expects a parent database call, but finds a  $\text{dep} = 0$  system call; or (iii) in response to Oracle Database bugs. When **mrprof** finds such a sequence of un-parented calls, it will synthesize a database call to act as the parent for the sequence. Although **mrprof** can unambiguously determine that such a parent call did in fact occur, it cannot determine the call's name, exact duration, or proportion of CPU duration to "unaccounted-for within dbcalls" duration. For each such event, **mrprof** will synthesize an

"uncategorizable within synthetic dbcalls" event.

In "Profile by Database Call" sections, **mrprof** may synthesize the following database calls in response to missing data in the input trace file:

between dbcalls

Represents the time spent between database calls. If the bulk of the response time has been consumed, for example, on the network, or in a Java program on an application tier, then this "between dbcalls" row will dominate your response time.

synthetic

A call that must have existed but was not recorded in the trace file. For example, a child database call should always be followed in the trace file by either a sibling call or its parent. If the following call is neither, or if the trace file EOF has been reached, then **mrprof** will synthesize a parent for the child.

synthetic misattributed

Synthesized in response to undesired Oracle Database behavior, in which the time attributed to statements by **mrprof** doesn't match the trace file duration. See *SYNTHETIC CALLS*/*uncategorizable within file* for the list of probable causes.

## EXAMPLES

```
$ mrprof ora_1492.trc
```

This command will create a Method R Profile report in a file called *ora\_1492.html* and open that file in your default browser.

```
$ mrprof --trcunit=.01 ora_1492.trc
```

This command will perform exactly as the prior command, except that it will pass the option **--trcunit=.01** to the **mrprof** kernel.

```
$ mrprof ora_*.zip
```

This command will create a Method R Profile report, in HTML format, for each file whose name matches the pattern **ora\_\*.zip** (interpreted by your OS shell), but it will not open them, unless the pattern matches only a single file name.

```
$ mrprof --open ora_*.zip
```

This command will create the same Method R Profile reports, and it will open them in your default browser.

## TROUBLESHOOTING

How can I test Java on my computer?

If you have Internet access, then visit <http://www.java.com/en/download/help/testvm.xml>. Or run the command **java -version** from a shell prompt. If the command succeeds, Java is visible from your path.

How can I tell what version of Java **mrprof** is using?

Run this:

```
$ mrprof --debug 2>&1 | grep -i java      # on Linux, macOS
> mrprof --debug 2>&1 | find /I "java"   &rem on Microsoft Windows
```

I don't have Java installed; how can I install it?

Visit <http://www.java.com/download>, and follow the instructions there.

I have more than one version of Java installed on my computer. How can I make **mrprof** use the right Java executable?

Put the directory of the Java executable that you want to use at the head of your **PATH**, or set the **JAVA\_HOME** environment variable to the parent of the bin directory that contains the Java executable that you want to use. **JAVA\_HOME** takes precedence over **PATH**.

How do I set my **JAVA\_HOME** or **PATH** environment variable?

**JAVA\_HOME** and **PATH** are just regular operating system environment variables. Set them either from your command shell or your system's preferences application.

How can I fix **java.lang.OutOfMemoryError** problems?

The Java Virtual Machine throws **OutOfMemoryError** when it cannot allocate enough memory. You can change **mrprof**'s JVM heap allocations with the **--jparam** command line option. Use **--jparam -Xms** to set the initial Java heap size, **--jparam -Xmx** (or **--max-java-heap**) to set the maximum Java heap size, and **--jparam -Xss** to set the Java thread stack size. For example, **--jparam -Xmx1900m** would set the maximum Java heap size to 1,900 megabytes.

How do I make my skew pictograms look like the ones at [method-r.com](http://method-r.com)?

Skew pictograms are supposed to resemble silhouettes of a city skyline. If yours look like a row of empty boxes or diamonds with question marks in them, then you are missing a font. Install either Profiler Pictogram (distributed in *\$MRWORKBENCH\_INSTALL/fonts/profilerpictogram.ttf*), or Arial Unicode MS.

## ENVIRONMENT VARIABLES

**mrprof** requires an environment variable called **MRWORKBENCH\_INSTALL** to be set so that the directory *\$MRWORKBENCH\_INSTALL/bin* contains the **mrprof** executable.

## EXIT STATUS

Exit status is 0 for success, 1 for failure.

## SUPPORT

For support, visit <http://method-r.com/support>.

## COPYRIGHT AND LICENSE

**mrprof** version 8.0.5.0

This is commercial software created and distributed by Method R Corporation.

Copyright 2001, 2018 Method R Corporation. All rights reserved.