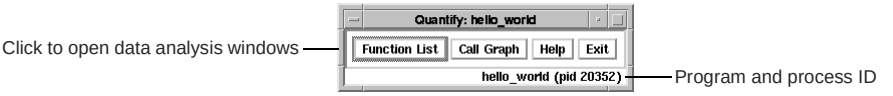
Rational® Quantify® Quick Reference

Using Quantify

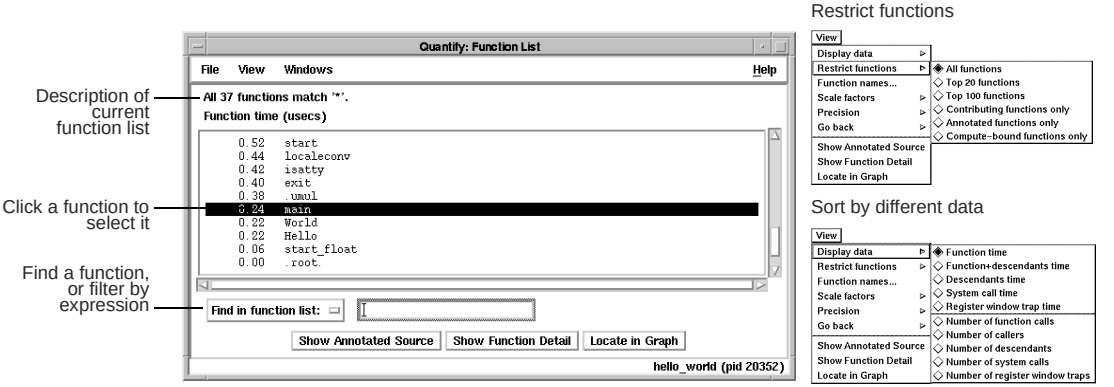
The CONTROL PANEL appears by default when running a Quantify'd program.
You can also invoke `qv` on a saved `.qv` file:

```
% qv a.out.23.0.qv
```

Control Panel: Open data analysis windows

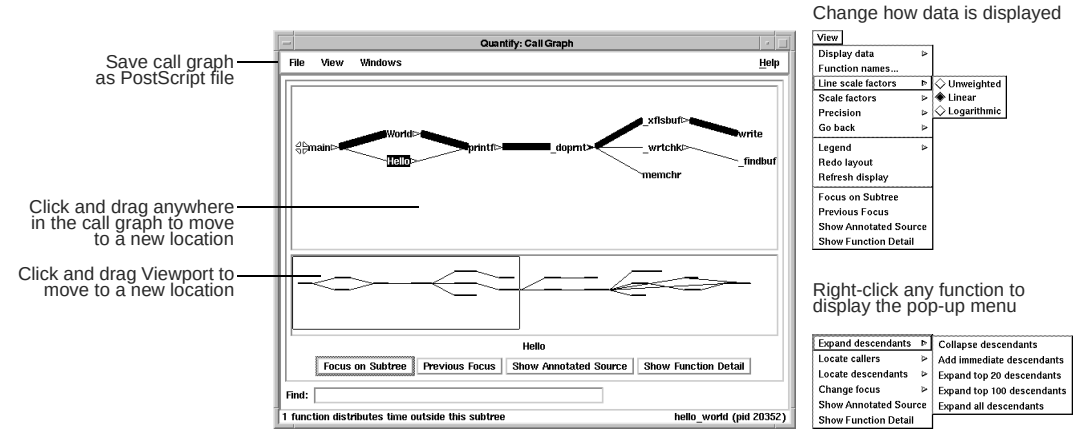


Function List window: Sort functions to find bottlenecks



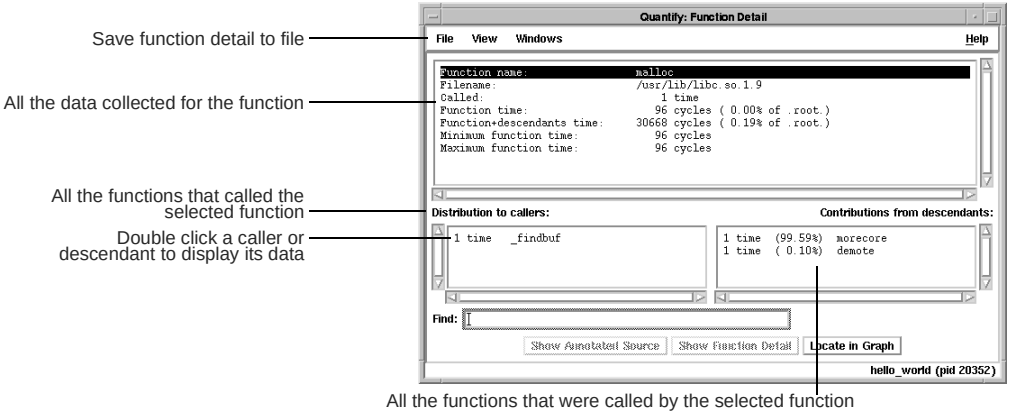
Call Graph window: Understand your application's calling structure

By default, Quantify expands the top 20 descendents of `.root`.



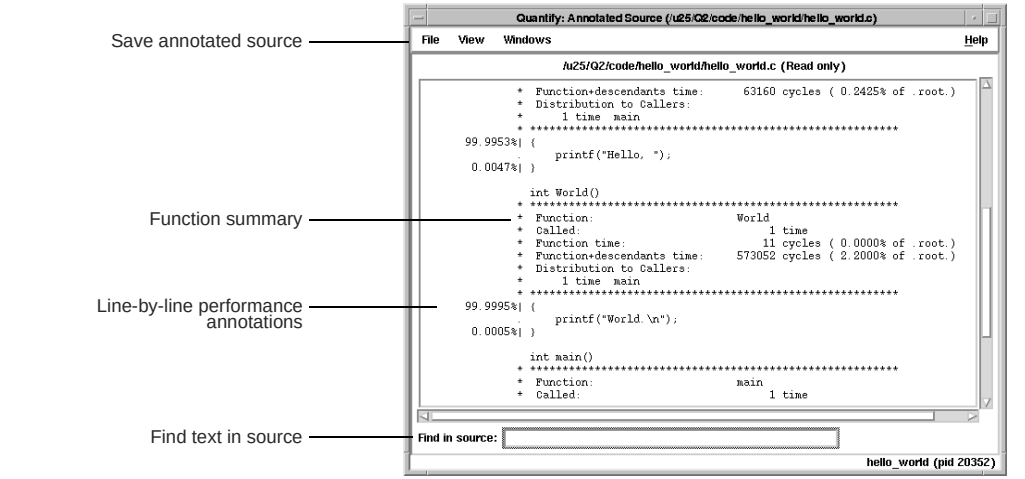
Rational® Quantify® Quick Reference

Function Detail window: Examine how a function's calling time is distributed



Annotated Source window: View line-by-line performance data

The Annotated Source window is available if you compile your program using the `-g` debugging option.



Conversion characters for filenames

Use these conversion characters when specifying filenames for options.

Character	Converts to	Character	Converts to
`%d`	Current date (yymmdd)	`%t`	Current time (hh:mm:ss)
`%p`	Process id (pid)	`%v`	Program name
`%n`	Sequence number, starting at 0. Increments each time the dataset is saved.	`%V`	Full pathname of the program with "/" replaced by "_"
`%T`	Thread identifier		

Rational® Quantify® Quick Reference

API functions

Include <quantifyhome>/quantify.h in your code and link with <quantifyhome>/quantify_stubs.a

Commonly used functions	Description
quantify_help (void)	Prints description of Quantify API functions
quantify_is_running (void)	Returns true if the executable is Quantify'd
quantify_print_recording_state (void)	Prints the recording state of the process
quantify_save_data (void)	Saves data from the start of the program or since last call to quantify_clear_data
quantify_save_data_to_file (char * filename)	Saves data to a file you specify
quantify_add_annotation (char * annotation)	Adds the specified string to the next saved dataset
quantify_clear_data (void)	Clears the performance data recorded to this point
quantify_<action>†_recording_data (void)	Starts and stops recording of all data
quantify_<action>†_recording_dynamic_library_data (void)	Starts and stops recording dynamic library data
quantify_<action>†_recording_register_window_traps (void)	Starts and stops recording register-window trap data
quantify_<action>†_recording_system_call (char *system_call_string)	Starts and stops recording specific system-call data
quantify_<action>†_recording_system_calls (void)	Starts and stops recording of all system-call data

† <action> is one of: start, stop, is. For example: quantify_stop_recording_system_call

Build-time options

Specify build-time options on the link line to build Quantify'd programs:

```
% quantify -cache-dir=$HOME/cache -always-use-cache-dir cc ...
```

Commonly used build-time options	Default
-always-use-cache-dir Specifies whether Quantify'd files are written to the global cache directory	no
-cache-dir Specifies the global cache directory	<quantifyhome>/cache
-collection-granularity Specifies the level of collection granularity	line
-collector Specifies the collect program to handle static constructors in C++ code	not set
-ignore-runtime-environment Prevents the run-time Quantify environment from overriding option values used in building the program	no
-linker Specifies an alternative linker to use instead of the system linker	system-dependent
-use-machine Specifies the build-time analysis of instruction times according to a particular machine	system-dependent

Rational® Quantify® Quick Reference

qv run-time options

To run `qv`, specify the option and the saved `.qv` file: `% qv -write-summary-file a.out.23.qv`

qv options	Default
`-add-annotation` Specifies a string to add to the binary file	not set
`-print-annotations` Writes the annotations to `stdout`	`no`
`-windows` Controls whether Quantify runs with the graphical interface	`yes`
`-write-export-file` Writes the recorded data in the dataset to a file in export format	not set
`-write-summary-file` Writes the program summary for the dataset to a file	not set

Run-time options

Specify run-time options using the environment variable `QUANTIFYOPTIONS`:

```
% setenv QUANTIFYOPTIONS "-windows=no"; a.out
```

Commonly used run-time options	Default
`-avoid-recording-system-calls` Avoids recording specified system calls	system-dependent
`-measure-timed-calls` Specifies measurement for timing system calls	`elapsed-time`
`-record-child-process-data` Records data for child processes created by `fork` and `vfork`	`no`
`-record-system-calls` Records system calls	`yes`
`-report-excluded-time` Reports time that was excluded from the dataset	`0.5`
`-run-at-exit` Specifies a shell script to run when the program exits	not set
`-run-at-save` Specifies a shell script to run each time the program saves counts	not set
`-save-data-on-signals` Saves data on fatal signals	`yes`
`-save-thread-data` Saves composite or per-stack thread data	`composite`
`-write-export-file` Writes the dataset to an export file as ASCII text	none
`-write-summary-file` Writes the program summary for the dataset to a file	`/dev/tty`
`-windows` Specifies whether Quantify runs with the graphical interface	`yes`