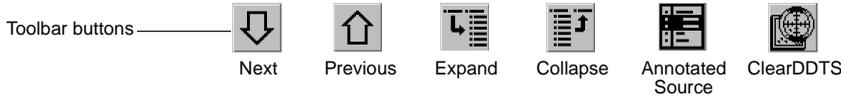


# Rational® PureCoverage® Quick Reference

## PureCoverage Viewer

Compile a program with the command `% purecov cc -g <filename>.c`

Run the program, then use the command `% purecov -view a.out.pcv` to open the Viewer.



Sorting order for summary rows

Summary rows

Adjusted	unused	lines	Runs	Calls	FUNCTIONS			ADJUSTED LINES			ADJS	total
					unused	used	used%	unused	used	used%		
Total Coverage												
/usr/home/pat/example/												
new_hello.c												
hello_world.c												
display_message												
main												
display_hello_world												

Number of runs over which data was collected for this file

Number of times this function was called

Statistics about functions

Statistics about lines

Number of adjusted lines

## PureCoverage Annotated Source Window

In the Viewer, click the button next to any function name to open the Annotated Source window.

Mark adjustments here: deadcode, inspected, tested

Number of times the line was executed

Source code with adjustment comments

Untested code

Adjusted code

Navigation and search region

Line	D	I	T	Hits	Annotated Source
16				1	display_hello_world();
17					else
18				0	display_message(argv[1]);
19				1	exit(0);
20				1	}
21					void
22					display_hello_world()
23					{
24					printf("Hello, World\n");
25				1	}
26				1	}
27					void
28					display_message(s)
29					{
30					char *s;
31					{
32		I			printf("%s, World\n", s); /* purecov: inspected */
33		I			/* purecov: inspected */
34					}

Next unused Prev unused Go to line #: Find:

/usr/home/pat/example/new\_hello.c

# Rational® PureCoverage® Quick Reference

## Build-time options

Set build-time options on the link line when you instrument programs with PureCoverage. For example:

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

Commonly used build-time options	Default
<b>-always-use-cache-dir</b> Forces all PureCoverage instrumented object files to be written to the global cache directory	<b>no</b>
<b>-auto-mount-prefix</b> Removes the prefix used by file system automounters	<b>/tmp_mnt</b>
<b>-cache-dir</b> Specifies the global directory where PureCoverage caches instrumented object files	<b>&lt;purecovhome&gt;/cache</b>
<b>-collector</b> Specifies the collect program to handle static constructors (for use with gcc, g++)	<b>none</b>
<b>-ignore-run-time-environment</b> Prevents the run-time PureCoverage environment from overriding the option values used in building the program	<b>no</b>
<b>-linker</b> Specifies a linker other than the system default for building the executables	<b>system-dependent</b>

## Run-time options

Set run-time options on the link line or by using the PURECOVOPTIONS environment variable. For example:

```
% setenv PURECOVOPTIONS "-counts-file=./test1.pcv `printenv PURECOVOPTIONS`"
```

Commonly used run-time options	Default
<b>-counts-file</b> Specifies an alternate file for writing coverage count data in binary format	<b>%v.pcv</b> <b>Note:</b> Can use filename conversion characters
<b>-follow-child-processes</b> Controls whether PureCoverage is enabled in forked child processes	<b>no</b>
<b>-log-file</b> Specifies a log file for PureCoverage run-time messages	<b>stderr</b> <b>Note:</b> Can use filename conversion characters
<b>-program-name</b> Specifies the full pathname of the PureCoverage instrumented program	<b>argv [0]</b>
<b>-user-path</b> Specifies a list of directories to search for source code	<b>none</b> <b>Note:</b> Can also be used in <b>-view</b> mode

## Analysis-time options

Use analysis-time options with analysis-time mode options, for example:

```
% purecov -merge=result.pcv -force-merge filea.pcv fileb.pcv
```

Commonly used analysis-time options	Default
<b>-apply-adjustments</b> Applies all adjustments in the \$HOME/.purecov.adjust file to exported coverage data	<b>yes</b>
<b>-force-merge</b> Forces the merging of coverage data files (.pcv files) obtained from different versions of the same object file	<b>no</b>

# Rational® PureCoverage® Quick Reference

## Analysis-time mode options

Command-line syntax: % purecov -<mode option> [analysis-time options] <file1.pcv file2.pcv ...>

Analysis-time mode options	Compatible options
<b>-export</b> Merges and writes coverage counts from multiple coverage data files (.pcv files) in export format to a specified file (-export=<filename>) or to stdout	<b>-apply-adjustments</b>
<b>-extract</b> Extracts adjustment data from source code files and writes it to \$HOME/.purecov.adjust	<b>none</b>
<b>-merge=&lt;filename.pcv&gt;</b> Merges and writes coverage counts from multiple coverage data files (.pcv files) in binary format	<b>-force-merge</b>
<b>-view</b> Opens the PureCoverage Viewer for analysis of one or more coverage data files (.pcv files)	<b>-force-merge, -user-path</b>

## Using PureCoverage with other Rational products

Product	Command line syntax
Purify	% purify <purifyoptions> purecov <purecovoptions> cc ...
Quantify	Cannot instrument for PureCoverage and Quantify simultaneously

## Conversion characters for filenames

Use these conversion characters when specifying filenames for options such as -log-file.

Character	Converts to
%v	Full pathname of program with "/" replaced by "_"
%n	Program name
%p	Process id (pid)
qualified filenames (./%v.plog)	Either absolute or relative to current working directory
unqualified filenames (no "/")	Directory containing the program

## Viewer keyboard accelerators

Key	Action	Menu equivalent
Control-a	Open Annotated Source window	Show annotated source in the Actions menu
Down arrow or Control-n	Move to the next block of uncovered code	Next in the Actions menu
Up arrow or Control-p	Move to the previous block of uncovered code	Previous in the Actions menu
Control-o	Expand the selected row	Expand in the Actions menu
Control-k or DEL	Collapse the selected row	Collapse in the Actions menu
Space or Return	Toggle the selected row between expanded and collapsed	Expand in the Actions menu Collapse in the Actions menu

# Rational® PureCoverage® Quick Reference

## Report scripts

Run report scripts from the command line, or choose **Run script** from the Viewer File menu to open the script dialog.



## Report scripts

<b>pc_annotate</b>	Produces an annotated source text file	
% pc_annotate [-force-merge] [-apply-adjustments=no] [-file=<basename>...] [-type=<type>] [<prog>.pcv...]		
<b>pc_below</b>	Reports low coverage	
% pc_below [-force-merge] [-apply-adjustments=no] [-percent=<pct>] [<prog>.pcv...]		
<b>pc_build_diff</b>	Compares PureCoverage data from two builds of an application	
% pc_build_diff [-apply-adjustments=no] [-prefix=XXXX...] old.pcv new.pcv		
<b>pc_covdiff</b>	Annotates the output of diff for modified source code	<b>Note:</b> Cannot be run from Viewer
% yourdiff <name>   pc_covdiff [-context=<lines>] [-format={diff side-by-side new-only}] [-lines=<boolean>] \ [-tabs=<stops>] [-width=<width>] [-force-merge] [-apply-adjustments=no] -file=<name> <prog>.pcv...		
<b>pc_diff</b>	Lists files for which coverage has changed	
% pc_diff [-apply-adjustments=no] old.pcv new.pcv		
<b>pc_email</b>	Mails a report to the last user who modified insufficiently covered files	
% pc_email [-force-merge] [-apply-adjustments=no] [-percent=<pct>] [<prog>.pcv...]		
<b>pc_select</b>	Identifies the subset of tests required to exercise modified source code	
% <list of changed files>   pc_select [-diff=<rules>] [-canonicalize=<rule>] test1.pcv test2.pcv...		
<b>pc_ssheet</b>	Produces a summary in spreadsheet format	
% pc_ssheet [-force-merge] [-apply-adjustments=no] [<prog>.pcv...]		
<b>pc_summary</b>	Produces an overall summary in table format	
% pc_summary [-file=<name>...] [-force-merge] [-apply-adjustments=no] [<prog>.pcv...]		

## API functions

Include `<purecovhome>/purecov.h` in your code and always link with `<purecovhome>/purecov_stubs.a`  
Useful compile or link options include: `-I'purecov -print-home-dir'` and `-L'purecov -print-home-dir'`

Function	Description
<code>int purecov_clear_data (void)</code>	Clears and resets all coverage accumulators to zero
<code>int purecov_disable_save (void)</code>	Prevents coverage counts from being written when the program exits or executes another program
<code>int purecov_enable_save (void)</code>	Permits coverage counts to be written when the program exits or executes another program
<code>int purecov_is_running (void)</code>	Returns 1 if the program is PureCoverage-instrumented
<code>int purecov_save_data (void)</code>	Merges and writes coverage counts accumulated before this function call
<code>int purecov_set_filename (const char *file_name)</code>	Sets the file for writing coverage data (default is <code>prog.pcv</code> ) <b>Note:</b> Can use filename conversion characters