



### Path

/opt/cadvise/bin/cadvise

### Mode of Operation

`cadvice [cadvise-options]user-compile-cmd [compiler-options]`

### Example Usage

Simple use of `cadvice` in conjunction with `user builds` - Only compile time checks are performed, no cross-file analysis is done.

```
% cadvice cc -c file1.c
% cadvice cc -c file2.c
```

Enable use of program database and do cross-file analysis. Use "report" mode of `cadvice` to analyze messages.

```
% cadvice -pdb mypdb cc -c file1.c
% cadvice -pdb mypdb cc -c file2.c
% cadvice -pdb mypdb cc file1.o file2.o
% cadvice report -pdb mypdb
```

### Types of Checks

`+wperfadvice [=1|2|3|4]`  
Enable performance advice.

`+wsecurity [=1|2|3|4]`  
Enable security vulnerability checking.

`+wlint`  
Enable lint style warnings.

`+w64bit`  
Enable 32-bit to 64-bit porting warnings.

`+wendian`  
Enable endianness checks.

`+wall`  
Enable `+wlint`, `+wsecurity`, `+w64bit` and `+wendian`.

`+wlock`  
Enable lock checks for programs using pthread based lock/unlock calls.

`+metrics [=ict] [:out=pdb|<filename>|+<filename>]`  
Emit program complexity metrics into the PDB or the file `filename`.

### Report Generation

```
cadvice report <report options> -pdb <pdb location>
cadvice report <report options> <log_file1> <log_file2>...
<log_fileN>
```

`-severity N`  
Print diagnostics with severity  $\geq N$ . Valid values for `N` are 1 - 9. Highest severity is 9 (Default value is 5).

`-summary`  
Print summary information about all diagnostics in the log files by severity level.

`-all`  
Print all diagnostics in the log files sorted by the severity level.

`-diag D1, D2, D3...Dn`  
Report messages with any of the diagnostics numbers `D1` to `Dn`.

`-file_summary`  
Print summary information for each source file.

`+metrics [=ict]`  
Print program complexity data for all or selected source files.

`-module <name>`  
Generate report only for source files that make up the module.

`-exclude <string1:string2...:stringn>`  
Suppress printing of diagnostics from source files that have any of the specified strings in the filename path.

`-include <string1:string2...:stringn>`  
Generate diagnostics from source files that contain specified strings in the file name path.

`-basepdb <basepdbname>`  
Generates a summary of differences between the PDB specified with `-pdb` option and the `basepdbname`.



invent

## HP Code Advisor C.02.05 (cadvise)

`-html[=dir]`  
Generate reports in HTML.

### Configurability

`+Warg1[, arg2, ..., argn]`  
Disable specific warning numbers `arg1 [, arg2,...argn]`

`-exclude <string1:string2...:stringn>`  
Suppress diagnostics from files that have any of the specified strings in the filename path. You can use this option for report generation.

`-include <string1:string2...:stringn>`  
Emit diagnostics from files that contain any of the specified strings in the filename path. You can use this option for report generation.

`+Wmacro:MACRONAME:d1,d2,d3,..dn`  
Disable warnings in a specified macro.

### Program Database (PDB)

`-pdb location`  
Specify the location (path) of the PDB.

`-snapshot`  
Freeze the PDB for future report comparisons.

`-pdb_destroy`  
Destroy PDB.

### Cross-file Analysis

`-crossfile=[none|pdb|defer|auto|module|list|listfile]`  
Control cross-file analysis in `cadvise`. Cross file analysis must be enabled to use the `+wsecurity`, `+wlock` and `+wall` options.

`-crossfile=none`  
Do not perform any cross-file analysis. This is the default option.

`-crossfile=pdb`  
Perform cross-file analysis for all files in the PDB

`-crossfile=defer`  
Do not perform cross-file analysis automatically. `cadvise` defers it until a later `-crossfile=[pdb|module|list|listfile]` run.

`-crossfile=auto`  
Perform cross-file analysis automatically when the user performs a link step. This is the default if PDB is specified.

`-crossfile=module:<module_name>`  
Enable cross-file analysis on the object files that make the binary.

`-crossfile=list:<file1:file2:...fileN>`  
Enable cross-file analysis on the specified object files.

`-crossfile=listfile:<list_file_name>`  
Enable cross-file analysis on all object files specified in the file, `list_file_name`.

### Other Options

`-compiler [aCC|cc|c89|c99|gcc|g++|ld]`  
Specify the compiler/linker command [`compiler-options`] used in `user-compile-cmd`. This allows mapping of the user command name to one of the standard commands recognized by `cadvise`.

`-[no]build`  
Do [not] invoke the user `compile/link` command. The default is `-build`. Automatic cross-file analysis is disabled with `-nobuild`.

`-[no]abort`  
Do [not] return a non-zero exit value for `cadvise` errors. The default is `-noabort`.

`+opts <filename>`  
Read options from file.

`-h|H|-help`  
Print a brief description and list of options for `cadvise`.