Name

tau_compiler.sh — Instrumenting source files.

Synopsis

`tau_compiler.sh` [`-p` *`profile`*] [`-optVerbose`] [`-optQuiet`] [`-optPdtDir=`*`dir`*] [`-optPdtF95Opts=`*`opts`*] [`-optPdtF95Reset=`*`opts`*] [`-optPdtCOpts=`*`opts`*] [`-optPdtCReset=`*`opts`*] [`-optPdtCxxOpts=`*`opts`*] [`-optPdtCReset=`*`opts`*] [`-optPdtF90Parser=`*`parser`*] [`-optGnuFortranParser`] [`-optGnuCleanscapeParser`] [`-optPdtUser=`*`opts`*] [`-optTauInstr=`*`path`*] [`-optDetectMemoryLeaks`] [`-optIncludeMemory`] [`-optTrackUPCR`] [`-optPreProcess`] [`-optCPP=`*`path`*] [`-optCPPOpts=`*`options`*] [`-optCPPReset=`*`options`*] [`-optTauSelectFile=`*`file`*] [`-optPDBFile=`*`file`*] [`-optTau=`*`opts`*] [`-optCompile=`*`opts`*] [`-optTauDefs=`*`opts`*] [`-optTauIncludes=`*`opts`*] [`-optReset=`*`opts`*] [`-optLinking=`*`opts`*] [`-optLinkReset=`*`opts`*] [`-optTauCC=`*`cc`*] [`-optOpariTool=`*`path/opari`*] [`-optOpariDir=`*`path`*] [`-optOpariOpts=`*`opts`*] [`-optOpariReset=`*`opts`*] [`-optOpari2Tool=`*`path/opari2`*] [`-optOpari2ConfigTool=`*`path/opari2_config`*] [`-optOpari2Dir=`*`path`*] [`-optOpari2Opts=`*`opts`*] [`-optOpari2Reset=`*`opts`*] [`-optNoMpi`] [`-optMpi`] [`-optNoRevert`] [`-optRevert`] [`-optKeepFiles`] [`-optAppC`] [`-optAppCXX`] [`-optAppF90`] [`-optShared`] [`-optCompInst`] [`-optPDTInst`] [`-optDisableHeaderInst`] { *`compiler`* } [*`compiler_options`*] [`-optTauWrapFile=`*`filename`*]

Description

The TAU Compiler provides a simple way to automatically instrument an entire project. The TAU Compiler can be used on C, C++, fixed form Fortran, and free form Fortran.

Options

`-optVerbose` Turn on verbose debugging messages.

`-optQuiet` Suppresses excessive output.

`-optDetectMemoryLeaks` Instructs TAU to detect any memory leaks in C/C++ programs.TAU then tracks the source location of the memory leak as wellas the place in the callstack where the memory allocation wasmade.

`-optPdtDir=<dir>` The PDT architecture directory. Typically `$(PDTDIR)/$(PDTARCHDIR)`.

`-optPdtF95Opts=<opts>` Options for Fortran parser in PDT (f95parse).

`-optPdtF95Reset=<opts>` Reset options to the Fortran parser to the given list.

`-optPdtCOpts=<opts>` Options for C parser in PDT (cparse). Typically `$(TAU_MPI_INCLUDE) $(TAU_INCLUDE) $(TAU_DEFS)`.

`-optPdtCReset=<opts>` Reset options to the C parser to the given list

`-optPdtCxxOpts=<opts>` Options for C++ parser in PDT (cxxparse). Typically `$(TAU_MPI_INCLUDE) $(TAU_INCLUDE) $(TAU_DEFS)`.

`-optPdtCxxReset=<opts>` Reset options to the C++ parser to the given list

`-optPdtF90Parser=<parser>` Specify a different Fortran parser. For e.g., `f90parse` instead of `f95parse`.

`-optGnuFortranParser=<parser>` Specify the GNU `gfortran` Fortran parser `gfparse`instead of `f95parse`

`-optGnuCleanscapeParser` Uses the Cleanscape Fortran parser `f95parse`instead of GNU's `gfparse`

`-optPdtUser=<opts>` Optional arguments for parsing source code.

`-optTauInstr=<path>` Specify location of tau_instrumentor. Typically `$(TAUROOT)/$(CONFIG_ARCH)/bin/tau_instrumentor`.

`-optIncludeMemory` Forinteral use only

`-optTrackUPCR` Adds tracking of the UPC runtime library.

`-optPreProcess` Preprocess the source code before parsing. Uses /usr/bin/cpp-P by default.

`-optCPP=<path>` Specify an alternative preprocessor and pre-process the sources.

`-optCPPOpts=<options>` Specify additional options to the C pre-processor.

`-optCPPReset=<options>` ResetC preprocessor options to the specified list.

`-optTauSelectFile=<file>` Specify selective instrumentation file for tau_instrumentor

`-optPDBFile=<file>` Specify PDB file for tau_instrumentor. Skips parsing stage.

`-optTau=<opts>` Specify options for tau_instrumentor.

`-optCompile=<opts>` Options passed to the compiler. Typically `$(TAU_MPI_INCLUDE) $(TAU_INCLUDE) $(TAU_DEFS)` .

`-optTauDefs=<opts>` Options passed to the compiler by TAU. Typically `$(TAU_DEFS)` .

`-optTauIncludes=<opts>` Options passed to the compiler by TAU. Typically `$(TAU_MPI_INCLUDE) $(TAU_INCLUDE)` .

`-optReset=<opts>` Reset options to the compiler to the given list

`-optLinking=<opts>` Options passed to the linker. Typically `$(TAU_MPI_FLIBS) $(TAU_LIBS) $(TAU_CXXLIBS)` .

`-optLinkReset=<opts>` Reset options to the linker to the given list.

`-optTauCC=<cc>` Specifies the C compiler used by TAU.

`-optOpariTool=<path/opari>` Specifies the location of the Opari tool.

`-optOpariDir=<path>` Specifies the location of the Opari directory.

`-optOpariOpts=<opts>` Specifies optional arguments to the Opari tool.

`-optOpariReset=<opts>` Resets options passed to the Opari tool.

`-optNoMpi` Removes `-l*mpi*` libraries during linking (default).

`-optMpi` Does not remove `-l*mpi*` libraries during linking.

`-optNoRevert` Exit on error. Does not revert to the original compilation rule on error.

`-optRevert` Revert to the original compilation rule on error (default).

`-optKeepFiles` Does not remove intermediate `.pdb` and `.inst.*` files.

`-optAppCC` Sets the failsafe C compiler.

`-optAppCXX` Sets the failsafe C++ compiler.

`-optAppF90` Sets the failsafe F90 compiler

`-optShared` Use shared library version of TAU

`-optCompInst` Use compiler-based instrumentation

`-optNoCompInst` Do not revert to compiler instrumentation if source instrumentation fails.

`-optPDTInst` Use PDT-based instrumentation

`-optHeaderInst` Enable instrumentation of headers

`-optDisableHeaderInst` Disable instrumentation of headers

`-optTrackIO` Specify wrapping of POSIX I/O calls at link time.

`-optWrappersDir=""` Specify the location of the link wrappers directory.

`-optTauUseCXXForC` Specifies the use of a C++ compiler for compiling C code

`-optTauWrapFile=<filename>` Specify path to the link_options.tau file generated by tau_wrap

`-optFixHashIf`
