RogueWave SOFTWARE

TotalView CLI コマンドガイド

一般的な CLI コマンド

alias: コマンドにユーザ定義の別名を付ける
alias *alias-name defn-body*
定義済みエイリアスの一覧
alias [*alias-name* **]**

capture: コマンド出力を文字列にキャプチャ
capture [–out | –err | –both]
[–f *filename* **]** *command*

dlappend: リスト要素を TotalView 変数に追加
dlappend *variable-name value* **[…]**

dset: CLI の状態変数を生成、変更、閲覧
dset *debugger-var value*
CLI の状態変数を閲覧
dset [*debugger-var* **]**
CLI の状態変数にデフォルト値を設定
dset -set_as_default *debugger-var valu*

dunset: 特定の CLI 変数をデフォルト値に戻す
dunset *debugger-var*
全ての CLI 変数をデフォルト変数に戻す
dunset –all

help: ヘルプ情報を表示する
help [*topic* **]**

stty: ターミナルのプロパティを設定
stty [*stty-args* **]**

unalias: 特定のエイリアスを削除
unalias *alias-name*
全てのエイリアスを削除
unalias -all

CLI の初期化と終了、プロセスを CLI に追加

dattach: プロセスにアタッチして管理下に置く
dattach [–g *gid* **] [–r** *hname* **] [–rank** *num* **]**
[–replay | –no_replay]
[–go | –halt]
[–ask_attach_parallel
| –no_attach_parallel]
[–c *corefile-name* **] [–e]** *fname*
[*pid-list* **]**
[–parallel_attach_subset
subset_specification **]**

ddetach: プロセスからデタッチする
ddetach

ddlopen: 共有ライブラリをロードする
ddlopen [–now | –lazy] [–local | –global]
[–mode *int* **]** *filespec*
共有ライブラリの情報を表示
ddlopen [–list *dll-ids...* **]**

dgroups: グループにメンバー追加
dgroups –add [–g *gid* **] [** *id-list* **]**
グループを消去
dgroups –delete [–g *gid* **]**
スレッド/プロセスとグループの共通部分を抜き出す
dgroups –intersect [–g *gid* **] [** *id-list* **]**
グループの情報を表示
dgroups [–list] [*pattern* **]**
新しいグループを作成
dgroups –new [*thread_or_process* **]**
[–g *gid* **] [** *id-list* **]**
グループからメンバーを除去
dgroups –remove [–g *gid* **] [** *id-list* **]**

dkill: プロセスの実行を終了
dkill [–remove]

dload: デバッグ情報をロード
dload [–g *gid* **] [–mpi** *starter* **] [–r** *hname***]**
[–replay | –no_replay]
[–env *variable=value* **]… [–e]**
executable
[–parallel_attach_subset
subset_specification **]**

drerun: プロセスをリスタート
drerun [*cmd_arguments* **] [<** *infile* **]**
[> [>][&] *outfile* **]**
[2> [>] *errfile***]**

drun: プロセスをスタート/リスタート
drun [*cmd_arguments* **] [<** *infile* **]**
[> [>][&] *outfile* **]**
[2> [>] *errfile***]**

dsession: セッションをロード
dsession [–load *session_name***]]**

exit: CLI セッションを終了
exit [–force]

quit: CLI セッションを終了
quit [–force]

プログラムの情報表示

dcalltree: パラレルバックトレースを表示
dcalltree [-data *pbv_data_array***]**
[–show_details] [–hide_backtrace]
[–sort columns] [–saveAsCSV
*filename***]**
[–saveAsDot *filename***]**

ddown: コールスタックを下に移動
ddown [num-levels]

dexamine: メモリの内容を表示
dexamine [–column_count *cnt* **]**
[–count *cnt* **] [–data_only]**
[–show_chars] [–string_length *len* **]**
[–format *fmt* **] [–memory_info]**
[–wordsize *size* **]**
variable_or_expression

dflush: 保留された expression evaluation の先頭を除去
dflush
全ての保留された *dprint* を除去
dflush –all
susp-eval-id までの保留された *dprint evaluations* を除去
dflush *susp-eval-id*

dga: *global array* 変数を表示
dga [–lang *lang_type* **]**
[*handle_or_name* **][** *slice* **]**

dlist: 現在の位置周辺にあるコードを表示
dlist [–n *num-lines* **]**
指定された位置周辺にあるコードを表示
dlist *source-loc* **[–n** *num-lines* **]**
現在の PC 周辺のコードを表示
dlist –e [–n *num-lines* **]**

dmstat: メモリ使用状況を表示
dmstat

dprint: 変数や expression の値を表示
dprint [–nowait] [–slice *slice_expr* **]**
[–stats [–data]]
variable_or_expression

dptsets: P/T expression の Tcl 配列内にあるプロセスやスレッドの状態を表示
dptsets [*ptset_array* **]**

dstatus: プロセス/スレッドの状態を表示
dstatus

dup: コールスタックを上に移動
dup [*num-levels* **]**

dwhat: 変数や関数名、シンボル情報を表示
dwhat *symbol-name*

dwhere: コールスタック内での位置を表示
dwhere [–level *num-levels* **] [** *num-levels* **]**
[–args] [–locals] [–registers]
[–show_image]] –noshow_pc]
[–noshow_fp]
全てのコールスタックを表示(デフォルト)
dwhere –all [–args] [–locals]
[–registers] [–show_image]
[–noshow_pc] [–noshow_fp]

実行を制御

dcont: 実行を再開し、停止するのを待つ
dcont

dfocus: 以降の CLI コマンドの対象を変更、または現在の値を返す
dfocus [*p/t-set* **]**
この P/T セットに対してコマンドを実行
dfocus *p/t-set command*

dgo: 対象のプロセスで実行を再開
dgo

dhalt: プロセスの実行を保留
dhalt

dhistory: ReplayEngine に関する操作
dhistory [–info] [–enable] [–disable]
[–create_bookmark [*comment***]]**
[–goto_bookmark *ID* **] [–go_live]**
[–show_bookmarks]
[–delete_bookmark *ID* **]**
Replay の履歴をファイルに保存
[–save [*recording-file***]]**
Deprecated なオプション
[–get_time] [–go_time *time* **]**

dhold: プロセスをホールド
dhold –process
スレッドをホールドする
dhold –thread

dnext: ソースをステップ実行し、関数をまたぐ
dnext [–back] [*num-steps* **]**

dnexti: アセンブリをステップ実行し、関数をまたぐ
dnexti [–replay | –no_replay] [*num-steps* **]**

dout: 現在の関数を呼び出した直後の位置まで実行
dout [–back] [*frame-count* **]**

dstep: ソースをステップ実行し、関数の中へ
dstep [–back] [*num-steps* **]**

dstepi: アセンブリをステップ実行し、関数の中へ

dstepi [–back] [*num-steps*]

dunhold: プロセスをリリース
dunhold –process
スレッドをリリース
dunhold –thread

duntil: 特定の位置まで実行
duntil [–back] *line-number*
あるアドレスまで実行
duntil [–back] –address *addr*
ある関数に入るまで実行
duntil *proc-name*

dwait: プロセス停止までコマンド入力をブロック
dwait

dworker: スレッドを worker group に追加/削除
dworker { *number* | *boolean* }

アクションポイント

dactions: アクションポイントについての情報を表示
dactions [*ap-id-list*] [–at *source-loc*]
[–enabled | –disabled]
[–enabled_blocks | –disabled_blocks]
[–block_images]
[–block_lines]
アクションポイントをファイルに保存
dactions –save [*filename*]
保存したアクションポイントをロード
dactions –load [*filename*]

dbarrier: バリアブレークポイントをある位置に作成
dbarrier *source -loc* [–pending]
[–stop_when_hit
{ group | process | none }]
[–stop_when_done
{ group | process | none }]
バリアブレークポイントをあるアドレスに作成
dbarrier –address *addr* [–pending]
[–stop_when_hit
{ group | process | none }]
[–stop_when_done
{ group | process | none }]

dbreak: ブレークポイントを作成
ソース内やアドレス内に
dbreak *loc* [–p | –g | –t]
[[–l *lang*] –e *expr*] [–pending]
dbreak –address *addr* [–p | –g | –t]
[[–l *lang*] –e *expr*]
[–pending]

ddelete: いくつかのアクションポイントを削除
ddelete *action-point-list*
全てのアクションポイントを削除
ddelete –a

ddisable: アクションポイントを有効化
ddisable *action-point-list*
ddisable –a

denable: アクションポイントを無効化
denable *action-point-list*
denable *–a*

dwatch: 変数やアドレスにウォッチポイントを割り当てる
dwatch *variable* [*–length byte-count*]
[–p | –g | –t]
[[–l *lang*] –e *expr*] [–t *type*]
dwatch –address *addr* –length *byte-count*
[–p | –g | –t]
[[–l *lang*] –e *expr*] [–t *type*]

一部のプラットフォーム用

dcuda: CUDA スレッドのデバッグを許可
dcuda { block [(x,y)] | thread [(x,y,z)] }
dcuda { kernel }
dcuda {device [<n>] | sm [<n>]
warp [<n>] | lane [<n>] }
dcuda { info-system | info-device
| info-sm | info-warp | info-lane }
dcuda { focus (Bx,By,Bz),(Tx,Ty)
| hwfocus <D/S/W/L> }

spurs: GDB の SPU Runtime System (SPU) ライブラリ使用時の情報を管理
変数 *OBJECT_SEARCH_PATH* にディレクトリ追加
spurs add [directory *directory-list ...* **]**

その他のコマンド

dassign: スカラー変数の値を変更
dassign *target value*

dcache: リモートライブラリのキャッシュを削除
dcache –flush

dcheckpoint: チェックポイントを作成
dcheckpoint [*after_checkpointing* **]**
SGI IRIX 上で
[–by *process_set*] [–no_park] [–force]
[–no_preserve_ids]
[–ask_attach_parallel
| –no_attach_parallel] *name*
IBM AIX 上で
dcheckpoint [*–delete* | *–halt* **]**

dheap: メモリデバッグ用のコマンド

drestart: チェックポイントをリスタート
AIX 上で
drestart [–halt] [–g *gid* **] [–r** *host* **]**
[–no_same_hosts]
SGI 上で
drestart [*process-state* **] [–no_unpark]**
[–g gid **] [–r** *host* **]**
[–ask_attach_parallel
| –no_attach_parallel]
[–no_preserve_ids] *checkpoint-name*

NameSpace コマンド

actionpoint: Actionpoint のプロパティ取得/設定
TV::actionpoint action [*obj-id* **] [** *other-args* **]**

dec2hex: 0 進数を 16 進数に変換
TV::dec2hex *number*
Reference Guide 参照

dll: 共有ライブラリの管理
TV::dll action [*dll-id-list* **] [** *other-args* **]**

errorCodes: 全てのエラーコードのタグを返す
TV::errorCodes
エラー情報を返すか raise する
TV::errorCodes *number_or_tag*
[–raise [*message* **]]**

expr: *dprint –nowait* の値を操作する
TV::expr action [*susp-eval-id* **] [** *other-args* **]**

focus_groups: フォーカス下のグループのリスト
TV::focus_groups

focus_processes: フォーカス下のプロセスのリスト
TV::focus_processes [–all | –group
| –process | –thread]

focus_threads: フォーカス下のスレッドのリスト
TV::focus_threads [–all | –group
| –process | –thread]

group: グループのプロパティを設定/取得
TV::group *action* **[** *object-id* **] [** *other-args* **]**

hex2dec: 10 進数に変換
TV::hex2dec *number*

process:プロセスのプロパティを設定/取得
TV::process *action* **[** *object-id* **] [** *other-args* **]**

read_symbols: シンボル読み込み
ライブラリから
TV::read_symbols –lib *lib-name-list*
特定のスタックフレームに関連したライブラリから
TV::read_symbols –frame [*number* **]**
バックトレース内の全フレームから
TV::read_symbols –stack

respond: コマンドへのレスポンスを提供
TV::respond *response command*

scope:内部スコープのプロパティを設定/取得
TV::scope *action* **[** *object-id* **] [** *other-args* **]**

source_process_startup: プロセスをロードする時に.tvd ファイルを"Sources"する
TV::source_process_startup *process_id*

symbol: TotalView のシンボル情報を取得/設定
TV::symbol *action* **[** *object-id* **] [** *other-args* **]**

thread:スレッドのプロパティを取得/設定
TV::thread *action* **[** *object-id* **] [** *other-args* **]**

type: 型のプロパティを取得/設定
TV::type *action* **[** *object-id* **] [** *other-args* **]**

type_transformation: type transformation を作成し、そのプロパティを examine する
TV::type_transformation *action*
[*object-id* **] [** *other-args* **]**