TotalView by Perforce

TotalView Cheat Sheet

1 Compiling Programs.

Compile your programs using the –g option. For example:

gcc -g –o *my_prog my_prog.c*

2 Starting TotalView.

Enter:

totalview *my_prog* **–a** *arguments*

Or, type **totalview** from the shell to open the Sessions Manager to:

- Start a new program or parallel program
- Attach to a running process
- Open a core file
- Manage your debug sessions

3 Toolbar Buttons Defined

- **Go** ( ): Starts execution.
- **Halt** ( ): Stops execution, but you can restart from where execution stopped.
- **Kill** ( ): Kills the executing program.
- **Restart** ( ): Does a **Delete**, then a **Go**.
- **Next** ( ): Executes all code on the current line; program counter (PC) will be at the next line.
- **Step** ( ): Executes line; if the line has a subroutine, PC moves into it.
- **Out** ( ): Executes remainder of current routine; PC is on the line that called this routine.
- **Run To** ( ): After selecting a line (click on the line, not the line number), press this button to execute all instructions from the PC until this line.
- **Modify Program Arguments** ( ): Launches the Session Editor where you can edit parameters or other aspects of your program.

4 Setting a Breakpoint

- **Line**: Click on a line number.
- **Function**: Select **Action Points > At Location**, then enter a breakpoint expression.
- **Search**: Select **Edit > Lookup File or Function**, then click on the line number.

5 Attaching to Already Running Programs

- Select **File > Attach to a Program** to launch the Sessions Manager, and browse to the program.
- If you don't see the program, use the **ps** command to find its PID (Program ID), then enter it in the **PID** field.

Always attach to a program's main thread.

6 Stopping at a Line Based on a Variable's Value

a. Set a Evaluation Point by either right-clicking on a line number and selecting **Create Evaluation Point** or by selecting **Action Points > Create Evaluation Point**.

b. Enter an expression to evaluate in the **Create Evaluation** dialog box, for example:

```
if (my_variable == 0) $stop
```

7 Seeing Variable Values

- For local variables, view the value in the **Local Variables** view's Value column or hover over the variable to view more info in a tooltip.
- For arrays and structures, either select [Add New Expression] in the **Data View** and enter the variable expression, or drag the variable from the Local Variables view to the Data View.

8 Chasing Pointer Values.

If a variable's type is a pointer, dereference it in the **Data View** to see its value.

9 Viewing Multiple Variables Concurrently.

Add any number of variables to the Data View, including struct and array elements. Watch the values update when you run your program.

10 Examine Array Statistics.

View stats generated from array values in the **Array Statistics** view by right-clicking on an array and selecting **Show Statistics**. Use the **Slice** field in the view to generate stats on a subset. Click **Update** to update the stats as your program runs.

11 Casting.

Change the way TotalView interprets and displays variable data by editing the **Type** field of a variable window.

For example, if you have a pointer to an array, you'll want to change the datatype from something like **int *** to **int[100] *** to see array or pointer elements.

12 Changing Variable Values.

In the **Local Variables** view and **Data View**, double-click a value to edit it.

13 STL Variables.

TotalView provides automatic STL type transformations to more clearly display STL data without the underlying structure. Control this default using the **TV::ttf** variable.

14 Searching For Variables or Other Program Elements.

Select **Edit > Find** (**Ctrl F**) to search for any program element.

15 Stopping Execution When a Variable's Value Changes.

Select **Action Points > Create Watchpoint.**

If the Data View is displaying an array or a structure, dive on an element so that only one of the variable's elements is displayed.

16 Seeing One Element in an Array of Structures as its own Array

a. Select one element of the array in the **Data View**.

b. Right-click and select **Dive in All**.

The Data View creates a new array to visualize this single element across structures.

17 Seeing a Variable's Value in Multiple Threads or Processes.

Select a thread or process in the **Processes & Threads** view. The Data View, Call Stack, and Local Variable view updates with data and variables relevant to the selection.

18 CLI Command Entry.

Select **Window > Views > Command Line**. Enter CLI commands in the displayed view. Enter **dhelp** to return help.

19 Debugging with fork() and execve() Programs.

In most cases, TotalView automatically follows fork() and execve() calls and acquires new processes into the debugging session. Use the **TV::exec_handling** and **TV::fork_handling** state variables to control how the debugger handles these system calls.

20 Debugging with ReplayEngine.

ReplayEngine is an add-on for reverse debugging in Linux x86 and x86-64. To enable reverse debugging, either:

- Select the checkbox **Enable reverse debugging with Replay Engine** in either the Debug a Program or Attach to Process dialogs in the Session Editor.
- Select **Debug > Enable ReplayEngine** after loading your program into TotalView, but before running it.

The ReplayEngine toolbar:

- **Record** ( ) Enables and disables ReplayEngine.
- **Go Back** ( ) Displays the state at the last action point. If no action point is found, displays the state at the start of its recorded history.
- **Prev** ( ) Displays the state at the previous statement execution. If that line had a function call, skips over the call.
- **Unstep** ( ) Displays the state at the previous statement execution. If that line had a function call, moves to the last statement in that function.
- **Caller** ( ) Displays the state before the current routine was called.
- **Back To** ( ) Displays the state for the line you select. This line must have executed prior to the currently displayed line.
- **Live** ( ) Shifts from replay mode to record mode and displays the statement that would have executed had you not moved into ReplayMode.
- **Bookmark** ( ) Creates a ReplayEngine bookmark at a selected location.
- **Save** ( ) Saves the current replay recording session to a file.

TotalView Complete Documentation
help.totalview.io

TotalView Video Tutorials
totalview.io/support/video-tutorials