Contents

New Features:
Versions 8.0 – 8.4.0

New Platforms and Compilers (8.0 through 8.4) ........................................... 1
New and Changed Features ........................................................................ 2
  Version 8.4 Features ........................................................................... 2
  Version 8.3 Features ......................................................................... 3
  Version 8.2 Features ........................................................................ 5
  Versions 8.0 and 8.1 Features ............................................................ 5
  Other Features .................................................................................. 7
New Features:
Versions 8.0 - 8.4.0

This document contains information about changes made to TotalView Debugger (TVD) for versions 8.0.0 and 8.4. While this information let you know what changes have occurred, it doesn’t completely describe these changes. Instead, you’ll find descriptions for most of the changes within the TotalView Debugger Users Guide and the TotalView Debugger Reference Guide.

TVD has many features and it gives you a great number of tools for finding your program’s problems. An easy way to get acquainted with these features is to subscribe to the “Tip of the Week”. If you subscribe to this mailing list, you’ll receive an email message every week (or most weeks) that tells you something about TVD and debugging.

- All of the tips are archived on our web site at http://www.totalviewtech.com/Documentation/Tips/index.html.
- If you like what you see, you can subscribe at http://www.totalviewtech.com/mojo/mojo.cgi.

New Platforms and Compilers (8.0 through 8.4)

8.4 Changes
TVD now supports Apple Mac OS X 11.4 (Leopard).

8.3 Changes
New operating system versions include:
- Apple OS X 10.4.5, 10.4.8, and 10.4.9
- Fedora Core 7
- Ubuntu 1.06

As always, we have added support for new versions of existing compilers and parallel runtime environments. Consult the TotalView Debugger Platforms and System Requirements Guide for more information.
8.2 Changes    TVD 8.2 has added support for the following systems and compilers:

- SiCortex Support
  SiCortex delivers ultra-low power, high-performance Linux computers for
  the HPC market. TVD will now support debugging parallel applications on
  the new SiCortex supercomputer.

- Cray XT4 Support and APLs Integration

- Fedora Core 6 Support

- Expanded Mac Support
  Preliminary Mac OS X Leopard Support, 64-bit Mac-Intel Support and
  Mac Universal Binaries Support

- Ubuntu Support
  Ubuntu is a community-developed Linux-based operating system for the
  desktop, laptop, thin client and server. TotalView will support applications
  developed on this new platform.

You’ll find a complete list of supported platforms and compilers in the
TotalView Debugger Platforms and System Requirements Guide.

New and Changed Features

Version 8.4 Features
This section lists the changes made for version 8.3 of TotalView Debugger
(TVD).

- If you have more than one TVD license, you can control which kind of li-
  cense TVD uses by adding one of the following command-line options:
  –team, –noteam, –teamplus, –noteamplus, –ent or –noent. For more in-
  formation, see Chapter 6 of the “TotalView Reference Guide.”

- The TVD Memory Debugger can now write light-weight memory debug
  files when an event occurs. These files are similar to the memory debug-
 ging files (.mdbg) files that you can write using the File > Export com-
  mand. They differ in that they are designed to be written when the event
  occurs and in such a way that the program’s behavior is minimally dis-
  turbed. These files are described in the Chapter 3 of the “Debugging Mem-
  ory Problems with TotalView Debugger” Guide.

- Improved support for C++ templates.

- Improved support for Fortran modules on Apple Mac OS X.
Version 8.3 Features

This section lists the changes made for version 8.3 of TotalView Debugger (TVD).

- Improvements to the way TVD launches MPI programs let you use TVD with virtually every MPI library, even with those that were not configured for debugging.
- TVD now highlights changes to values displayed in the **Variable** and **Expression List** windows with a colored background. (See Figure 1 on page 3.)

![Figure 1: Highlighted Change in the Variable Window](image)

While this figure shows a simple variable, TVD also highlights changed elements within compound variables such as structures and arrays.

- Values in the **Variable** and **Expression List** windows have a *Previous value* hidden column that you can display. Use the control on the right side of the column headings to display a list of columns that you can display or hide. (See Figure 2 on page 3.)

![Figure 2: Last Value Column](image)
New and Changed Features

- When a process hits a breakpoint, TVD highlights the breakpoint by placing an arrow over the breakpoint ID in the Action Points pane. (See Figure 3 on page 4.)

![Figure 3: Highlighted Action Point ID](image)

- View > Show Across replaces View > Laminate in the Variable window's menus. This means the commands you will now use are View > Show Across > Process and View > Show Across > Thread.

- You can now tell TVD to show a variable across processes or threads by right-clicking on it in the Source pane, then selecting either Across Processes or Across Threads from the context menu.

- The Create Watchpoint command was added to the Action Points menu. As always, you can create a watchpoint from within the Variable window by selecting Tools > Create Watchpoint.

- You can now set a watchpoint upon a variable's memory address by right-clicking on the variable in the Source pane and then selecting Create Watchpoint from the context menu.

- You can completely expand or collapse information in the Variable window by selecting an icon in the toolbar. The accelerators for these commands are Ctrl++ (that’s the control key and the + symbol) and Ctrl+- (which is the control key and the - symbol).

- TVD no longer stops by default when your program loads a library.

- You can specify more than one core file on the command line and you can use wildcards in core file names.

- There is a new Events View report within the Memory Debugger Heap Status tab.

![Figure 4: Event View](image)
Within the Memory Debugger’s File > Import Data dialog box, you can select multiple memory debugging (.mdbg) files.

**Version 8.2 Features**

This section looks at changes that have occurred within TVD.

- **Early-Access GUI Installer**
  You can now install TVD from tar files as you’ve always done or install it using our new graphical installer. We are calling this an early-access release in that we want you to tell us what you think of it and how we can improve it.

- **Fortran Parameter Display**
  TVD now displays the value of Fortran parameters. Parameters can be used like variables in expressions but could not previously be examined within the debugger.

**Versions 8.0 and 8.1 Features**

**Breakpoint Changes**

Action points are considerably more powerful. Here is a summary:

- The Action Point > At Location command now has three choices:
New and Changed Features

- **Function or Line**: lets you set a line number or a function name. This choice is what occurred in previous TVD releases.
- **All Methods in Class**: lets you set a breakpoint on all methods in a class. This can set more than one breakpoint.
- **All Virtual Functions and Overrides**: lets you set breakpoints on virtual functions and their overrides. This too can set more than one breakpoint.

You can now tell TVD that a breakpoint will occur in a library that will be loaded later and that TVD should retain knowledge of this breakpoint. This allows it to be set when the library is read. Previously, TVD had to create and set breakpoints at the same time. This meant that when you enter a name into the **At Location** dialog box and the name is not yet known, TVD displays either an **Ambiguous Function** or a **Question** dialog box. At this time, you can set the breakpoint’s status to **pending**. (See Figure 6 on page 6.)

When you create a barrier breakpoint or change a breakpoint to a barrier point, the same new features are available.

The CLI **dbreak**, **dbarrier**, and **dlist** commands have been extended to use these features. The argument to these commands can now be a break-
point expression. Understanding this concept reveals some of the sub-
tleties involved using these new features. These concepts are explained
with the dbreak pages of the TotalView Debugger Reference Guide.

Other Features
This section describes improvements and changes made in many different
places in TVD.

- The way in which you set search paths has changed. (See Figure 7 on
  page 7.)

You have told us that most of the time, you usually do not need to set
search paths. If you do, you just need to enter a couple of paths. This is
what the old dialog box was designed for. (You can still do this by typing
paths directly into the EXECUTABLE_PATH tab.) However, when your pro-
grams make the transition from modules being developed on your work-
station to an place where the work of development teams is brought
together, setting search paths was tedious and difficult to get right. This
new dialog box, extensively documented in the help, lets you solve this
problem.

- The File > New Program dialog box has changed. Much of what you will
  see makes the dialog box more usable. Most notable is the way you at-
tach to processes. The dialog box now lets you select more than one
process at a time. The one new feature is that you can now enable mem-
ory debugging from this dialog box.
The **View > Freeze** command is added to the Variable window. This command tells TVD that it should freeze the contents of a Variable window. That is, as your program executes and as data values change, the contents of this window does not change. In most cases, you will also create a second Variable Window so that you can see old and new values at the same time.

The **View > Lock** command is added to the Variable Window. This command tells TVD that it should not change the address from which the Variable Window is obtaining information.

New **dheap --compare** CLI command options. This options lets you compare the result of two different memory states.

New **dkill --remove** CLI command option. Using this option to tell TVD that, in addition killing the process, it should remove knowledge of the process. This is seldom necessary. However, if you are using TotalView Team Debugger, using this option makes the token used by a process available to another process in your program.

The following CLI variables were added for this release

- **TV::env**: sets an environment variable.
- **TV::bluegene_server_launch_string**: sets the Blue Gene server launch string.
- **TV::default_stderr_append**: tells TVD to append **stderr** information to **stdout**.
- **TV::default_stderr_filename**: tells TVD to write **stderr** information to a file.
- **TV::default_uds_filename**: tells TVD to write **stderr** information to a file.
- **TV::default_stdout_append**: tells TVD to append **stdout** information to a file.
- `TV::default_stdout_filename`: Tells TVD to write `stdout` information to a file.