

These release notes contain a summary of new features and enhancements, late-breaking product issues, migration from earlier releases, and bug fixes.

PLEASE NOTE: The version of this document in the product distribution is a snapshot at the time the product distribution was created. Additional information may be added after that time because of issues found during distribution testing or after the product is released. To be sure you have the most up-to-date information, see the version of this document on the Rogue Wave web site:

<http://www.roguewave.com/support/product-documentation/totalview.aspx>

Additions and Updates

Significant scalability and performance improvements

This release focuses on a new scalability infrastructure on Linux, Blue Gene/Q, and Cray platforms that uses efficient broadcast and reduction operations for a quantum leap in scalability. TotalView can now be used across hundreds of thousands of processes and millions of threads. This change is particularly noticeable in the reduction in time between when you launch a job and when you can begin actual debugging.

Improved MemoryScape startup performance

Scalability, performance, and reduced startup times have also been implemented for memory debugging. At very large scales, focusing on a single process is likely to result in the greatest improvements.

Root window aggregate display improvements

The root window now displays an aggregated tree of information about the state of the processes and threads you are debugging, rather than the former list-based display. The new display offers flexibility in the type of data that is aggregated, and maintains the full ability to dive on processes and threads.

Remote Display Client

Version 1.4.0 of the Remote Display Client is now available. RDC v1.4.0 includes a new look and enhancements including supporting additional Linux versions, ensuring profile data is saved properly, aligning correctly on various sized displays and other minor fixes.

Compiler Changes

The following new compilers are now supported by TotalView:

- Intel compiler 15.0
- PGI C++ 14.4

Bug Fixes for 8.15

- TVT-15436 The TotalView Mac version should use the dSYM files for debug information instead of requiring the .o files to be saved
- TVT-15864 RDC does not work on Ubuntu 12.10 and 13
- TVT-17116 Fatal Error when shifting between the Action Point Address dialogs of two different Boost types
- TVT-17472 Fatal Error on Cray: register_raw_library: Library 'xxxx' appeared with different asect counts: 2 != 0
- TVT-17640 Update FlexIm to version 11.12.1-2 or higher in order to support customers with large numbers of devices on their Macs
- TVT-17701 Internal error parsing a null string; message: "Encountered unsupported location operator"
- TVT-17743 Fatal Error when diving on member of unordered_set<int*>
- TVT-17748 TotalView indirect launch does not work for SGI's MPT
- TVT-17980 Various issues with unordered map/set/multimap/multiset transformations for different combinations of compilers and Boost library versions
- TVT-18167 TotalView GUI shows incorrect results for an unordered multiset example with gcc headers greater than 4.7.2

Known Issues

Licensing

Trouble executing licensing scripts on Linux hosts due to a new FlexNet version and LSB 3 compliance

When running the new licensing software (flexlm-11.11 or flexlm-11.12) the scripts may not be able to run the lmhostid or lmgrd commands. This version of FlexNet Publisher software requires a run-time linker with a file name that denotes Linux Standard Base 3 compliance. This requirement can be met by installing the "lsb-3" (or higher) package for the host where the licensing software resides or simply creating the appropriately named symbolic link (e.g. /lib/ld-lsb.so.3, /lib64/ld-lsb-x86-64.so.3, or /lib64/ld-lsb-ppc64.so.3).

TotalView sometimes cannot acquire license due to FlexNet bug

If you are using Linux Power, Linux Itanium, or AIX then you are still affected by the bug in the FlexNet Publisher software that results in TotalView not being able to acquire a license when your license file contains multiple licenses with different maintenance expiry dates (i.e. the 4th field on the INCREMENT line). The licensing software seems to skip over some of the licenses in this case. After ascertaining that your license file is being read and is correct, and you think you might be running into this bug, we recommend that you add the "sort" keyword and value (such as sort=1, sort=2, sort=3) to each INCREMENT line in the license file in any order. This bug has been reported to Flexera and is identified as SIOC-000145042.

Here is an example of adding the "sort" keyword:

```
SERVER linux-power 0050569b402c
VENDOR toolworks
INCREMENT TotalView_Enterprise toolworks 2014.1231 permanent 1 \
    sort=1 VENDOR_STRING="processors=16 platform=linux-power" \
    SIGN=3350A26C395A
INCREMENT TotalView_Enterprise toolworks 2015.1231 permanent 1 \
    sort=2 VENDOR_STRING="processors=16 platform=linux-ia64" \
    SIGN=C7D11FB667C8
INCREMENT TotalView_Enterprise toolworks 2016.1231 permanent 1 \
    sort=3 VENDOR_STRING="processors=16 platform=linux-x86_64" \
    SIGN=BEC7534A248A
```

Mac OS X

With multiple displays attached to a Mac OS X machine, some TotalView windows may not be noticeable

When a user attaches an external monitor to a running Macbook Pro, or adds multiple displays to a Mac, window rearrangement may move some windows off screen. This may result in a TotalView modal window not being found until one uses the Mac command to display all the windows (Mission Control). This appears to be an interaction between XQuartz and Darwin. It has been seen in Mavericks, but it's possible it will show up in other releases. There may be a workaround in System Preferences->Mission Control by disabling "Displays have separate Spaces"

Physical console access needed when starting TotalView

Starting in Mountain Lion, OS X security policies require that users meet a password challenge in order to use TotalView, and the challenge can only be issued to the console (the OS X desktop). After the password challenge is met once, TotalView can be run repeatedly from the same login session without further challenges.

It is possible to work around this need for physical access with the following steps. The first few of these are likely already set in order to allow TotalView to run.

- Install XQuartz and TotalView
- Ensure every user needing debugging is in the `_developer` group
- Allow X11 forwarding in the `sshd_config` file (disabled by default)
- In a terminal window enter the following two commands:
`DevToolsSecurity -enable` (this step is optional if this was already enabled)
`sudo security authorizationdb write system.privilege.taskport allow`

TotalView Remote Display Client hangs

If you are using the Remote Display Client on Mac OS X Mountain Lion or Mavericks the application may freeze if more than one Terminal window is open. To get around this issue set the display number in the Terminal's Advanced Options dialog to a value that will not conflict with another user.

Yosemite crash

There has been at least one report of an intermittent crash of the Mac itself while debugging an MPI program with TotalView on the first release of Yosemite. Crash reports have been sent to Apple as this is symptomatic of a kernel bug.

Ubuntu

[ptrace_scope setting causes various failures with TotalView and MemoryScape](#)

Beginning with Ubuntu 10.10, the kernel places a security restriction that prevents attaching to a process that was not directly started by the current process. This security restriction prevents TotalView from debugging applications that are not directly started by the debugger, which would include most MPI jobs or currently running processes. You can turn off this feature by entering:

- `su root // a simple sudo of the following does not seem to work`
- `echo 0 > /proc/sys/kernel/yama/ptrace_scope`

This will enable debugging and attaching to processes. This setting will last until the system is rebooted. For a permanent change, change the `kernel.yama.ptrace_scope` setting on the last line in `/etc/sysctl.d/10-ptrace.conf`.

[Memory debugging by linking against the TotalView libraries may not work](#)

There are a number of cases where it is recommended to link the Heap Interposition Agent (HIA) into the target program to allow memory debugging without having to enable it in the GUI each time. Starting in Ubuntu 12, the linker does not link in libraries that are not directly used by the program. This means that the link line for the agent, with `TVLIB` pointing to the TotalView library,

```
gcc -g -o memprog memprog.c -L$TVLIB -ltvheap -Wl,-rpath,$TVLIB
```

may not pick up the HIA. Instead, one must add the option `"-Wl,--no-as-needed"` before the inclusion of the `tvheap` library. The new compile/link line will look like

```
gcc -g -o memprog memprog.c -Wl,--no-as-needed -L$TVLIB -ltvheap -Wl,-rpath,$TVLIB
```

CUDA

Dynamic parallelism not fully supported

With CUDA 5.5-6.5 we have limited support for dynamic parallelism. You should be able to use TotalView in the CUDA 5.5/CUDA 6.5 runtime with applications that display dynamic parallelism; however we plan improvements to our functionality for displaying the relationships between dynamically launched kernels and navigating the various running kernels.

Layered textures not supported

TotalView does not yet support CUDA layered textures. If you try to examine a layered texture in the TotalView Data Pane, a "Bad address" message will be displayed and you will see "ERROR: Reading Texture memory not currently supported" displayed on the console. If you require layered textures support, please contact TotalView support at support@roguewave.com and let us know how you are using textures so we can develop the best solution to support you.

SGI

Memory debugging MPI programs on SGI systems needs special linking

The TotalView and MemoryScape memory debugging Heap Interposition Agent (HIA) technology conflicts with the SGI memory manager when used in MPI programs. The easiest way to get around this problem is to disable the SGI memory manager by unsetting the MPI_MEM_ALIGN environment variable. Without this variable set, the SGI memory manager will not be loaded and the HIA will work correctly, enabling memory debugging to take place.

Cray

Debugging your program within supercomputer environments can often be challenging. Reference the sections below to learn pointers on how to use Cray esLogin Nodes, successfully enable memory debugging, and perform reverse debugging on your program within a Cray environment.

esLogin Nodes

If you are using an esLogin node on the Cray, attempts to start TotalView on aprun from the esLogin nodes will not work. TotalView will start up, but when control is transferred to the compute nodes, TotalView will end up focused on ssh, rather than aprun itself. On the esLogin node, aprun is really an esWrap function which starts up an ssh session on the compute node which then runs aprun. TotalView will pick up the ssh program, but will not be able to follow it as it invokes the real aprun. We have a preliminary solution for this issue. If you are interested in trying it out for your Cray environment please contact TotalView support.

Memory debugging on Cray systems

Use the pointers below to help achieve a successful memory debugging session within the Cray environment:

- **Install TotalView on a shared file system visible to the Cray compute nodes**
In order for the required memory debugging shared libraries to be located when your program is running on a compute node it is best to install TotalView on a shared file system.
- **Cray TotalView Support Module is not required for TotalView 8.15.0**
As of TotalView 8.15.0 and its use of the MRNet tree framework TotalView no longer requires the Cray TotalView Support Library to be installed in order to run. If the MRNet tree framework is turned off then the Cray TotalView Support Module will be required.
- **Statically linking your program against the tvheap_cnl library**
One of the most foolproof ways of enabling memory debugging is to statically link against the tvheap_cnl library that is shipped with TotalView. The static library fully supports multithreaded applications. Statically linking your application with the tvheap_cnl library will automatically enable memory debugging in your program when debugged under TotalView and MemoryScape. See the "Linking Your Application with

the Agent” discussion in the User Guide for more information on how to statically link your applications with the library.

- **Dynamically linking your program against the tvheap_cnl library**

It is possible to dynamically link your application against the dynamic version of the tvheap_cnl library. In this scenario the tvheap_cnl library must be visible to the Cray Compute Node systems either through a shared file system or by the Cray environment automatically staging the applications shared library dependencies on the compute node.

- **Cray Compute Node /tmp directory must be writable**

Currently, the memory debugging shared library, tvheap_cnl, requires that /tmp be writable and it does not use the environment variable TMPDIR in order to use a different temporary directory. These issues are being tracked in TVT-18267 and will be fixed for the next TotalView release.

- **Do not enable memory debugging on the aprun starter process**

Turning on memory debugging for the aprun starter process will cause it to fail and prevent the job from starting. The proper way to enable memory debugging is to use the static or dynamic linking options described above.

Reverse debugging on Cray systems

Use the pointers below to help achieve a successful reverse debugging session within the Cray environment:

- **Do not enable reverse debugging on the aprun starter process**

Turning on reverse debugging for the aprun starter process will cause it to fail and prevent the job from starting. The proper way to turn on reverse debugging is to launch your parallel job and reach a breakpoint in the code and then dynamically turn on the Replay Engine reverse debugging option. It will begin recording the execution of the program from that point forward.