

TOTALVIEW

**COMMAND LINE
SYNTAX**



VERSION 8.6



Copyright © 2007–2008 by TotalView Technologies. All rights reserved

Copyright © 1998–2007 by Etnus LLC. All rights reserved.

Copyright © 1996–1998 by Dolphin Interconnect Solutions, Inc.

Copyright © 1993–1996 by BBN Systems and Technologies, a division of BBN Corporation.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without the prior written permission of TotalView Technologies.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

TotalView Technologies has prepared this manual for the exclusive use of its customers, personnel, and licensees. The information in this manual is subject to change without notice, and should not be construed as a commitment by TotalView Technologies. TotalView Technologies assumes no responsibility for any errors that appear in this document.

TotalView and TotalView Technologies are registered trademarks of TotalView Technologies.

TotalView uses a modified version of the Microline widget library. Under the terms of its license, you are entitled to use these modifications. The source code is available at <http://www.totalviewtech.com/Products/TotalView/developers>.

All other brand names are the trademarks of their respective holders.

Contents



1	TotalView Command Syntax	
	Command-Line Syntax	1
	Command-line Options	2
2	TotalView Debugger Server Command Syntax	
	The tvdsvr Command and Options.....	11
	Options	12
	Replacement Characters	15
	INDEX	19

TotalView Command Syntax



This chapter describes the syntax of the **totalview** command. Topics in this chapter are:

- Command-Line Syntax
- Command-line Options

Command-Line Syntax

- Format:* **totalview** [*filename* [*corefile*]] [*options*]
- Arguments:*
- | | |
|-----------------|---|
| <i>filename</i> | Specifies the path name of the executable being debugged. This can be an absolute or relative path name. The executable must be compiled with debugging symbols turned on, normally the -g compiler option. Any multiprocess programs that call fork() , vfork() , or execve() should be linked with the dbfork library. |
| <i>corefile</i> | Specifies the name of a core file. Use this argument in addition to <i>filename</i> when you want to examine a core file with TotalView. |
- Description:* TotalView is a source-level debugger with a motif-based graphic user interface and features for debugging distributed programs, multiprocess programs, and multithreaded programs. TotalView is available on a number of different platforms.
- If you specify mutually exclusive options on the same command line (for example, **-dynamic** and **-no_dynamic**), the last option listed is used.

Command-line Options

- a** *args* Pass all subsequent arguments (specified by *args*) to the program specified by *filename*. This option must be the last one on the command line.
- aix_use_fast_trap** Tells TotalView that it support the AIX fast trap mechanism. You must either set this option on the command line or place it within a **.tvdrc** file. That is, you cannot interactively set it using the CLI.
- background** *color* Sets the general background color to *color*.
default: light blue
- bg** *color* Same as **-background**.
- compiler_vars** (Alpha, HP, and SGI only.) Shows variables created by the Fortran compiler, as well as those in the user's program.
Some Fortran compilers (HP f90/f77, HP f90, SGI 7.2 compilers) output debugging information that describes variables the compiler itself has invented for purposes such as passing the length of character*(*) variables. By default, TotalView suppresses the display of these compiler-generated variables.
However, you can specify the **-compiler_vars** option to display these variables. This is useful when you are looking for a corruption of a run-time descriptor or are writing a compiler.
- no_compiler_vars** (Default) Tells TotalView that it should not show variables created by the Fortran compiler.
- control_c_quick_shutdown**
- ccq** (Default) Tells TotalView to kills attached processes and exits.
- no_control_c_quick_shutdown**
- nccq** Invokes code that sometimes allows TotalView to better manage the way it kills parallel jobs when it works with management systems. This has only been tested with SLURM. It may not work with other systems.
- dbfork** (Default) Catches the **fork()**, **vfork()**, and **execve()** system calls if your executable is linked with the **dbfork** library.
- no_dbfork** Tells TotalView that it should not catch **fork()**, **vfork()**, and **execve()** system calls even if your executable is linked with the **dbfork** library.

- `-debug_file consoleoutputfile`
 Redirects TotalView console output to a file named *consoleoutputfile*.
- default:* All TotalView console output is written to **stderr**.
- `-demangler=compiler`
 Overrides the demangler and mangler TotalView uses by default. The following indicate override options.
- | | |
|---------------------------------|-------------------------|
| <code>-demangler=compaq</code> | HP cxx on Linux (alpha) |
| <code>-demangler=dec</code> | HP Tru64 C++ or Fortran |
| <code>-demangler=gnu</code> | GNU C++ on Linux Alpha |
| <code>-demangler=gnu_dot</code> | GNU C++ on Linux x86 |
| <code>-demangler=gnu_v3</code> | GNU C++ Linux x86 |
| <code>-demangler=hp</code> | HP aCC compiler |
| <code>-demangler=irix</code> | SGI IRIX C++ |
| <code>-demangler=kai</code> | KAI C++ |
| <code>-demangler=kai3_n</code> | KAI C++ version 3.n |
| <code>-demangler=kai_4_0</code> | KAI C++ |
| <code>-demangler=spro</code> | SunPro C++ 4.0 or 4.2 |
| <code>-demangler=spro5</code> | SunPro C++ 5.0 or later |
| <code>-demangler=sun</code> | Sun CFRONT C++ |
| <code>-demangler=xlc</code> | IBM XLC/VAC++ compilers |
- `-display displayname`
 Set the name of the X Windows display to *displayname*. For example, `-display vinnie:0.0` will display TotalView on the machine named "vinnie."
- default:* The value of your **DISPLAY** environment variable.
- `-dll_ignore_prefix list`
 The colon-separated argument to this option tells TotalView that it should ignore files having this prefix when making a decision to ask about stopping the process when it *dlopens* a dynamic library. If the DLL being opened has any of the entries on this list as a prefix, the question is not asked.
- `-dll_stop_suffix list`
 The colon-separated argument to this option tells TotalView that if the library being opened has any of the entries on this list as a suffix, it should ask if it should open the library.
- `-dpvm`
 HP Tru64 UNIX *only*: Enable support for debugging the HP Tru64 UNIX implementation of Parallel Virtual Machine (PVM) applications.

- dump_core** Allows TotalView to dump a core file of itself when an internal error occurs. This is used to help TotalView Technologies debug problems.
- e *commands*** Tells TotalView to immediately execute the CLI commands named within this argument. All information you enter here is sent directly to the CLI's Tcl interpreter. For example, the following writes a string to **stdout**:
- ```
cli -e 'puts hello'
```
- You can have more than one **-e** option on a command line.
- ent** Tells TotalView that it should only use an Enterprise license.
- no\_ent** Tells TotalView that it should not use an Enterprise license. You may combine this with **-no\_team** or **-noteamplus**.
- env *variable=value*** Tells TotalView to add an environment variable to the environment variables passed to your program by the shell. If the variable already exists, it effectively replaces the previous value. You need to use this command for each variable being added; that is, you cannot add more than one variable with an **env** command.
- foreground *color*** Sets the general foreground color (that is, the text color) to *color*.
- default:* **black**
- fg *color*** Same as **-foreground**.
- f9x\_demangler=*compiler*** Overrides the Fortran demangler and mangler TotalView uses by default. The following indicate override options.
- ```
-demangler=sprow_f9x_4    SunPro Fortran, 4.0 or later
-demangler=xlf            IBM Fortran
```
- global_types** (Default) Lets TotalView assume that type names are globally unique within a program and that all type definitions with the same name are identical. The C++ standard asserts that this must be true for standard-conforming code.
- If this option is set, TotalView will attempt to replace an opaque type (**struct foo *p;**) declared in one module, with an identically named defined type in a different module.
- If TotalView has read the symbols for the module containing the non-opaque type definition, then when displaying variables declared with the opaque type, TotalView will automatically display the variable by using the non-opaque type definition.

- no_global_types**
 Specifies that TotalView *cannot* assume that type names are globally unique in a program. You should specify this option if your code has multiple different definitions of the same named type, since otherwise TotalView can use the wrong definition for an opaque type.
- gnu_debuglink** Tells TotalView that if a program or library has a **.gnu_debug_link** section, it should look for a **gnu_debug_link** file. If found, TotalView reads the debugging information from this file.
- no_gnu_debuglink**
 Do not load information from a **gnu_debug_link** file even if the file has a **.gnu_debug_link** section.
- gnu_debuglink_checksum**
 Tells TotalView that it should validate the **gnu_debug_link** file's checksum against the checksum contained in the process's **.gnu_debuglink** section.
- no_gnu_debuglink_checksum**
 Do not compare checksums. Only do this if you are absolutely certain that the debug file matches.
- ipv6_support** Directs TotalView to support IPv6 addresses.
- no_ipv6_support**
 (Default) Do not support IPv6 addresses.
- kcc_classes** (Default) Converts structure definitions output by the KCC compiler into classes that show base classes and virtual base classes in the same way as other C++ compilers. See the description of the **TV::kcc_classes** variable for a description of the conversions that TotalView performs.
- no_kcc_classes**
 Specifies that TotalView will not convert structure definitions output by the KCC compiler into classes. Virtual bases will show up as pointers, rather than as data.
- lb** (Default) Loads action points automatically from the *file-name.TVD.v3breakpoints* file, providing the file exists.
- nlb** Tells TotalView that it should not automatically load action points from an action points file.
- local_interface *string***
 Sets the interface name that the server uses when it makes a callback. For example, on an IBM PS2 machine, you would set this to `css0`. However, you can use any legal **inet** interface name. (You can obtain a list of the interfaces if you use the **netstat -i** command.)
- memory_debugging**
 Enables memory debugging. By adding the following suboptions, you enable that particular feature using its the feature's default configuration. In most cases, you will want to use one or more of the following suboptions.

- mem_notify_events**
Turns on memory event notification.
 - no_mem_notify_events** turns event notification off.
 - mem_guard_blocks**
Surrounds allocated memory blocks with guard blocks.
 - mem_hoard_freed_memory**
Tells the Memory Debugger to hoard memory blocks instead of releasing them when a **free()** routine is called.
 - mem_paint_on_alloc**
Paint memory blocks with a bit pattern when they are allocated.
 - mem_paint_on_dealloc**
Paint memory blocks with a bit pattern when they are freed.
 - mem_paint_all**
Paint both allocated and deallocated blocks with a bit pattern.
- message_queue**
(Default) Enables the display of MPI message queues when debugging an MPI program.
- mqd** Same as **-message_queue**.
- no_message_queue**
Disables the display of MPI message queues when you are debugging an MPI program. This might be useful if something is overwriting the message queues and causing TotalView to become confused.
- no_mqd**
Same as **-no_message_queue**.
- mpi starter** Names the MPI that your program requires. The list of starter names that you enter are those that appear in the **Parallel system** pulldown list contained within the **New Program**'s Parallel tab. If the starter name has more than one word (for example, **Open MPI**), enclose the name in quotes. For example:
-mpi "Open MPI"
- nodes** Specifies the number of nodes upon which the MPI job will run.
- np** Specifies how many tasks that TotalView should launch for the job. This argument usually follows a **-mpi** command-line option.
- nptl_threads** Tells TotalView that your application is using NPTL threads. You only need use this option if default cannot determine that you are using this threads package.
- no_nptl_threads**
Tells TotalView that you are not using the NPTL threads package. Use this option if TotalView thinks your application is using it and it isn't.

- parallel** (Default) Enables handling of parallel program run-time libraries such as MPI, PE, and UPC.
- no_parallel** Disables handling of parallel program run-time libraries such as MPI, PE, and UPC. This is useful for debugging parallel programs as if they were single-process programs.
- patch_area_base** *address* Allocates the patch space dynamically at *address*. See "Allocating Patch Space for Compiled Expressions" in Chapter 15 of the *TotalView Users Guide*.
- patch_area_length** *length* Sets the length of the dynamically allocated patch space to this *length*. See "Allocating Patch Space for Compiled Expressions" in the *TotalView Users Guide*.
- pid** *pid* Tells TotalView to attach to process *pid* after it starts executing.
- procs** Specifies how many tasks that TotalView should launch for the job. This argument usually follows a **-mpi** command-line option.
- pvm** Enables support for debugging the ORNL implementation of Parallel Virtual Machine (PVM) applications.
- remote** *hostname[:portnumber]* Debugs an executable that is not running on the same machine as TotalView. For *hostname*, you can specify a TCP/IP host name (such as **vinnie**) or a TCP/IP address (such as **128.89.0.16**). Optionally, you can specify a TCP/IP port number for *portnumber*, such as **:4174**. When you specify a port number, you disable the autolaunch feature. For more information on the autolaunch feature, see "Setting Single Process Server Launch" in the *TotalView Users Guide*.
- r** *hostname[:portnumber]* Same as **-remote**.
- replay** Tells TotalView that it should enable the ReplayEngine when TotalView begins. This command-line option is ignored if you do not have a license for ReplayEngine.
- s** *pathname* Specifies the path name of a startup file that will be loaded and executed. This path name can be either an absolute or relative name.
You can add more than one **-s** option on a command line.
- serial** *device[:options]* Debugs an executable that is not running on the same machine as TotalView. For *device*, specify the device name of a serial line, such as **/dev/com1**. Currently, the only *option* you are allowed to specify is the baud rate, which defaults to **38400**. For more information on debugging

over a serial line, see “*Debugging Over a Serial Line*” in Chapter 4 of the *TotalView Users Guide*.

`-search_path pathlist`

Specify a colon-separated list of directories that TotalView will search when it looks for source files. For example:

```
totalview -search_path proj/bin:proj/util
```

`-signal_handling_mode "action_list"`

Modifies the way in which TotalView handles signals. You must enclose the *action_list* string in quotation marks to protect it from the shell.

An *action_list* consists of a list of *signal_action* descriptions separated by spaces:

```
signal_action[ signal_action] ...
```

A signal action description consists of an action, an equal sign (=), and a list of signals:

```
action=signal_list
```

An *action* can be one of the following: **Error**, **Stop**, **Resend**, or **Discard**. For more information on the meaning of each action, see Chapter 3 of the *TotalView Users Guide*.

A *signal_specifier* can be a signal name (such as **SIGSEGV**), a signal number (such as 11), or a star (*), which specifies all signals. We recommend that you use the signal name rather than the number because number assignments vary across UNIX sessions.

The following rules apply when you are specifying an *action_list*:

- (1) If you specify an action for a signal in an *action_list*, TotalView changes the default action for that signal.
- (2) If you do not specify a signal in the *action_list*, TotalView does not change its default action for the signal.
- (3) If you specify a signal that does not exist for the platform, TotalView ignores it.
- (4) If you specify an action for a signal more than once, TotalView uses the last action specified.

If you need to revert the settings for signal handling to built-in defaults, use the **Defaults** button in the **File > Signals** dialog box.

For example, here’s how to set the default action for the **SIGTERM** signal to resend:

```
“Resend=SIGTERM”
```

Here's how to set the action for **SIGSEGV** and **SIGBUS** to error, the action for **SIGHUP** to resend, and all remaining signals to stop:

"Stop=* Error=SIGSEGV,SIGBUS Resend=SIGHUP"

- shm "action_list"** Same as **-signal_handling_mode**.
- starter_args "arguments"** Tells TotalView to pass *arguments* to the starter program. You can omit the quotation marks if *arguments* is just one string without any embedded spaces.
- stderr pathname** Names the file to which your program will write **stderr** information while executing within TotalView. If the file exists, TotalView overwrites it.
- stderr_append** Tells TotalView to append the information if writes to **stderr** to the file named in the **-stderr** command or the file named in GUI or a TotalView variable. If the file does not exist, TotalView creates it.
- stderr_is_stdout pathname** Tells TotalView to redirect **stderr** to **stdout**.
- stdin pathname** Names the file from which your program will read information while executing within TotalView.
- stdout pathname** Names the file to which your program will write **stdout** information while executing within TotalView. If the file exists, TotalView overwrites it.
- stdout_append** Tells TotalView to append the information if writes to **stdout** to the file named in the **-stdout** command or the file named in GUI or a TotalView variable. If the file does not exist, TotalView creates it.
- tasks** Specifies how many tasks that TotalView should launch for the job. This argument usually follows a **-mpi** command-line option.
- team** Tells TotalView that it should only use a Team license.
- no_team** Tells TotalView that it should not use an Enterprise license. You may combine this with **-no_ent** or **-noteam-plus**.
- teampplus** Tells TotalView that it should only use a Team Plus license.
- no_templplus** Tells TotalView that it should not use a Team Plus license. You may combine this with **-no_ent** or **-noteam**.
- tvhome pathname** The directory from which TotalView reads preferences and other related information and the directory to which it writes this information.

Command-line Options

- user_threads** (Default) Enables handling of user-level (M:N) thread packages on systems where two-level (kernel and user) thread scheduling is supported.
- no_user_threads** Disables handling of user-level (M:N) thread packages. This option may be useful in situations where you need to debug kernel-level threads, but in most cases, this option is of little use on systems where two-level thread scheduling is used.
- xterm_name** *pathname* Sets the name of the program used when TotalView needs to create a the CLI. If you do not use this command or have not set the **TV::xterm_name** variable, TotalView will attempt to create an **xterm** window.
- verbosity** *level* Sets the verbosity level of TotalView messages to *level*, which may be one of **silent**, **error**, **warning**, or **info**.
default: info

TotalView Debugger Server

Command Syntax



This chapter summarizes the syntax of the TotalView Debugger Server command, **tvdsvr**, which is used for remote debugging. Remote debugger occurs when you explicitly call for it or when you are using disciplines like MPI that startup processes on remote servers. For more information on remote debugging, refer to “Setting Up Remote Debugging Sessions” in the *TotalView Users Guide*.

Topics in this chapter are:

- The **tvdsvr** Command and Its Options
- Replacement Characters

The **tvdsvr** Command and Options

Format: `tvdsvr {-server | -callback hostname:port | -serial device} [other options]`

Description: **tvdsvr** allows TotalView to control and debug a program on a remote machine. To accomplish this, the **tvdsvr** program must run on the remote machine, and it must have access to the executables being debugged. These executables must have the same absolute path name as the executable that TotalView is debugging, or the **PATH** environment variable for **tvdsvr** must include the directories containing the executables.

You must specify a **-server**, **-callback**, or **-serial** option with the **tvdsvr** command. By default, TotalView automatically launches **tvdsvr** using the **-callback** option, and the server establishes a connection with TotalView. (Automatically launching the server is called *autolaunching*.)

If you prefer not to automatically launch the server, you can start **tvdsvr** manually and specify the **-server** option. Be sure to note the password that **tvdsvr** prints out with the message:

```
pw = hexnumhigh:hexnumlow
```

TotalView will prompt you for *hexnumhigh:hexnumlow* later. By default, **tvdsvr** automatically generates a password that it uses when establishing connections. If desired, you can set your own password by using the **-set_pw** option.

To connect to the **tvdsvr** from TotalView, you use the **File > New Program** Dialog Box and must specify the host name and TCP/IP port number, *hostname:portnumber* on which **tvdsvr** is running. Then, TotalView prompts you for the password for **tvdsvr**.

Options

The following options name the port numbers and passwords that TotalView uses to connect with **tvdsvr**.

-callback *hostname:port*

(Autolaunch feature only) Immediately establishes a connection with a TotalView process running on *hostname* and listening on *port*, where *hostname* is either a host name or TCP/IP address. If **tvdsvr** cannot connect with TotalView, it exits.

If you use the **-port**, **-search_port**, or **-server** options with this option, **tvdsvr** ignores them.

-callback_host *hostname*

Names the host upon which the callback is made. The *hostname* argument indicates the machine upon which TotalView is running. This option is most often used with a bulk launch.

-callback_ports *port-list*

Names the ports on the host machines that are used for callbacks. The *port-list* argument contains a comma-separated list of the host names and TCP/IP port numbers (*hostname:port,hostname:port...*) on which TotalView is listening for connections from **tvdsvr**. This option is most often used with a bulk launch.

For more information, see Chapter 4, "Setting Up Remote Debugging Sessions" in the *TotalView Users Guide*.

-debug_file *console_outputfile*

Redirects TotalView Debugger Server console output to a file named *console_outputfile*.

Default: All console output is written to **stderr**.

- dpvm** Uses the HP Tru64 UNIX implementation of the Parallel Virtual Machine (DPVM) library process as its input channel and registers itself as the DPVM tasker. This option is not intended for users launching tvdsvr manually. When you enable DPVM support within TotalView, TotalView automatically uses this option when it launches tvdsvr.
- nodes_allowed *num*** Explicitly tells tvdsvr how many nodes the server supports and how many licenses it needs. This is only used for the Cray XT3.
- port *number*** Sets the TCP/IP port number on which tvdsvr should communicate with TotalView. If this port is busy, tvdsvr does not select an alternate port number (that is, it won't communicate with anything) unless you also specify **-search_port**.
- Default:** 4142
- pvm** Uses the ORNL implementation of the Parallel Virtual Machine (PVM) library process as its input channel and registers itself as the ORNL PVM tasker. This option is not intended for users launching tvdsvr manually. When you enable PVM support within TotalView, TotalView automatically uses this option when it launches tvdsvr.
- search_port** Searches for an available TCP/IP port number, beginning with the default port (4142) or the port set with the **-port** option and continuing until one is found. When the port number is set, tvdsvr displays the chosen port number with the following message:
- port = *number***
- Be sure that you remember this port number, since you will need it when you are connecting to this server from TotalView.
- serial *device*[:*options*]** Waits for a serial line connection from TotalView. For *device*, specifies the device name of a serial line, such as **/dev/com1**. The only *option* you can specify is the baud rate, which defaults to **38400**. For more information on debugging over a serial line, see "Debugging Over a Serial Line" in Chapter 4 of the TotalView Users Guide.
- server** Listens for and accepts network connections on port 4142 (default). Using **-server** can be a security problem. Consequently, you must explicitly enable this feature by placing an empty file named **tvdsvr.conf** in your **/etc** directory. This file must be owned by user ID 0 (root). When tvdsvr

encounters this option, it checks if this file exists. This file's contents are ignored.

You can use a different port by using one of the following options: `-search_port` or `-port`. To stop `tvdsvr` from listening and accepting network connections, you must terminate it by pressing Ctrl+C in the terminal window from which it was started or by using the `kill` command.

`-set_pw hexnumhigh:hexnumlow`

Sets the password to the 64-bit number specified by the *hexnumhigh* and *hexnumlow* 32-bit numbers. When a connection is established between `tvdsvr` and TotalView, the 64-bit password passed by TotalView must match this password set with this option. `tvdsvr` displays the selected number in the following message:

```
pw = hexnumhigh:hexnumlow
```

We recommend using this option to avoid connections by other users.

If necessary, you can disable password checking by specifying the `"-set_pw 0:0"` option with the `tvdsvr` command. Disabling password checking is dangerous; it allows anyone to connect to your server and start programs, including shell commands, using your UID. Therefore, we do not recommend disabling password checking.

`-set_pws password-list`

Sets 64-bit passwords. TotalView must supply these passwords when `tvdsvr` establishes the connection with it. The argument to this command is a comma-separated list of passwords that TotalView automatically generates. This option is most often used with a bulk launch.

For more information, see Chapter 4, "Setting Up Remote Debugging Sessions" in the *TotalView Users Guide*.

`-verbosity level`

Sets the verbosity level of TotalView Debugger Server-generated messages to *level*, which may be one of **silent**, **error**, **warning**, or **info**.

Default: info

`-working_directory directory`

Makes *directory* the directory to which TotalView connects.

Note that the command assumes that the host machine and the target machine mount identical file systems. That is, the path name of the directory to which TotalView is connected must be identical on both the host and target machines.

After performing this operation, the TotalView Debugger Server is started.

Replacement Characters

When placing a **tvdsvr** command in a **Server Launch** or **Bulk Launch** string (see the **File > Preferences** command within the online Help for more information), you will need to use special replacement characters. When your program needs to launch a remote process, TotalView replaces these command characters with what they represent. Here are the replacement characters:

%B	Expands to the bin directory where tvdsvr is installed.
%C	Is replaced by the name of the server launch command being used. On most platforms, this is rsh . On HP, this command is remsh . If the TVDSVRLAUNCHCMD environment variable exists, TotalView will use its value instead of its platform-specific value.
%D	Is replaced by the absolute path name of the directory to which TotalView will be connected.
%F	Contains the "tracer configuration flags" that need to be sent to tvdsvr processes. These are system-specific startup options that the tvdsvr process needs.
%H	Expands to the host name of the machine upon which TotalView is running. (This replacement character is most often used in bulk server launch commands. However, it can be used in a regular server launch and within a tvdsvr command contained within a temporary file.)
%I	Expands to the pid of the MPI starter process. For example, it can contain mpirun , aprun , etc. It can also be the process to which you manually attach. If no pid is available, %I expands to 0.
%J	Expands to the job ID. For MPICH or poe jobs, is the contents of the totalview_jobid variable contained either in the starter or first process. If that variable does not exist, it is set to zero ("0"). If it is not appropriate for the kind of job being launched, its value is -1.
%K	(Red Storm and BlueGene architectures) If TotalView must use an alternative name for tvdsvr , specify its name here. For example, on BlueGene, the server name is tvdsvr_bgf . On Red Storm, it is tvdsvr_rs .
%L	If TotalView is launching one process, this is replaced by the host name and TCP/IP port number (<i>hostname:port</i>) on which TotalView is listening for connections from tvdsvr . If a bulk launch is being performed, TotalView replaces this with a comma-separated list of the host names and

	TCP/IP port numbers (<i>hostname:port,hostname:port...</i>) on which TotalView is listening for connections from tvdsvr . For more information, see Chapter 4, “ <i>Setting Up Remote Debugging Sessions</i> ” in the <i>TotalView Users Guide</i> .
%M	(Sun) Expands to the command name used for a local server launch.
%N	Is replaced by the number of servers that TotalView will launch. This is only used in a bulk server launch command.
%P	If TotalView is launching one process, this is replaced by the password that it automatically generated. If a bulk launch is being performed, TotalView replaces this with a comma-separated list of 64-bit passwords.
%R	Is replaced by the host name of the remote machine specified in the File > New Program command. When performing a bulk launch, this is replaced by a comma-separated list of the names of the hosts upon which TotalView will launch tvdsvr processes.
%S	If TotalView is launching one process, it replaces this symbol with the port number on the machine upon which TotalView is running. If a bulk server launch is being performed, TotalView replaces this with a comma-separated list of port numbers.
%t1 and %t2	Is replaced by files that TotalView creates containing information it generates. This is only available in a bulk launch. These temporary files have the following structure: (1) An optional header line containing initialization commands required by your system. (2) One line for each host being connected to, containing host-specific information. (3) An optional trailer line containing information needed by your system to terminate the temporary file. The File > Preferences Bulk Server Page allows you to define templates for the contents of temporary files. These files may use these replacement characters. The %N , %t1 , and %t2 replacement characters can only be used within header and trailer lines of temporary files. All other characters can be used in header or trailer lines or within a host line defining the command that initiates a single-process server launch. In header or trailer lines, they behave as defined for a bulk launch within the host line. Otherwise, they behave as defined for a single-server launch.
%U	(Sun) Expands to the local socket ID.

<code>%V</code>	Is replaced by the current TotalView verbosity setting.
<code>%Z</code>	Expands to the job ID. For MPICH or poe jobs, is the contents of the totalview_jobid variable contained either in the starter or first process. If that variable does not exist, it is set to zero ("0"). If it is not appropriate for the kind of job being launched, its value is -1.

Index



–serial device 13

Symbols

%B server launch replacement character 15
%C server launch replacement character 15
%D path name replacement character 15
%H hostname replacement character 15
%L host and port replacement character 15
%N line number replacement character 16
%P password replacement character 16
%S source file replacement character 16
%t1 file replacement character 16
%t2 file replacement character 16
%V verbosity setting replacement character 17

A

–a option to totalview command 2
action points
 loading automatically 5
aix_use_fast_trap command-line option 2
arguments
 for totalview command 1
 for tvdsrv command 12

B

background command-line option 2
baud rate, specifying 13
bg command-line option 2
bulk launch 15

C

C++
 demangler 3
callback command-line option 11, 12
callback_host 12
callback_ports 12
ccq command-line option 2
CLI commands
 executing immediately 4
color
 foreground 4
comand-line options
 –ccq 2
command-line options
 –aix_use_fast_trap 2
 –background 2
 –bg 2
 –ccq 2
 –compiler_vars 2
 –control_c_quick_shutdown 2
 –dbfork 2
 –debug_file 3
 –demangler 3
 –display 3
 –dll_ignore_prefix 3
 –dll_stop_suffix 3
 –dpvm 3
 –dump_core 4
 –e 4
 –ent 4
 –env 4
 –f9x_demangler 4
 –fg 4
 –foreground 4
 –global_types 4
 –gnu_debuglink 5
 –gnu_debuglink_checksum 5
 –ipv6_support 5
 –kcc_classes 5
 –lb 5

–message_queue 5, 6
–mqd 6
–nccq 2
–nlb 5
–no_compiler_vars 2
–no_control_c_quick_shutdown 2
–no_dbfork 2
–no_ent 4
–no_global_types 5
–no_gnu_debuglink 5
–no_gnu_debuglink_checksum 5
–no_ipv6_support 5
–no_kcc_classes 5
–no_message_queue 6
–no_mqd 6
–no_nptl_threads 6
–no_parallel 7
–no_team 9
–no_teamplus 9
–no_user_threads 10
–nptl_threads 6
–parallel 7
–patch_area_base 7
–patch_area_length 7
–pid 7
–r 7
–remote 7
–s 7
–search_path 8
–serial 7
–shm 9
–signal_handling_mode 8
–stderr 9
–stderr_append 9
–stderr_is_stdout 9
–stdin 9
–stdout 9
–stdout_append 9
–team 9
–timeplus 9
–tvhome 9

D

- user_threads 10
- verbosity 10
- xterm_name 10
- commands
 - totalview 1
 - tvdsrv, syntax and use 11
- compiler_vars command-line option 2
- connection directory 15
- console output for tvdsrv 12
- console output redirection 3
- control_c_quick_shutdown command-line option 2
- core
 - dumping for TotalView 4

D

- dbfork command-line option 2
- dbfork library
 - syntax 2
- debug_file command-line option 3, 12
- debugger server 11
- demangler
 - overriding 3, 4
- demangler command-line option 3
- display command-line option 3
- dll_ignore_prefix command-line option 3
- dll_stop_suffix command-line option 3
- dpvm command-line option 3, 13
- dump_core command-line option 4

E

- e command-line option 4
- enabling PVM support 7
- ent command-line option 4
- env command-line option 4
- execve()
 - calling 2

F

- f9x_demangler command-line option 4
- fg command-line option 4
- foreground command-line option 4
- fork()
 - calling 2

G

- global_types, command-line option 4
- gnu_debuglink command-line option 5
- gnu_debuglink_checksum command-line option 5

H

- handling signals 8
- host ports 12
- hostname
 - expansion 15
 - for tvdsrv 12

- replacement 16

I

- ignoring libraries by prefix 3
- inet interface name 5
- interface name for server 5
- ipv6_support command-line option 5
- ipv6_support option 5

K

- kcc_classes command-line option 5
- kcc_classes command-line option 5

L

- launching
 - tvdsrv 11
- lb command-line option 5
- libraries
 - dbfork 2
 - ignoring by prefix 3
- loading
 - action points 5

M

- mangler, overriding 3, 4
- message_queue command-line option 5, 6
- mqd command-line option 6

N

- naming the host 12
- nccq command-line option 2
- nlb command-line option 5
- no_compiler_vars command-line option 2
- no_control_c_quick_shutdown command-line option 2
- no_dbfork command-line option 2
- no_ent command-line option 4
- no_global_types command-line option 5
- no_gnu_debuglink command-line option 5
- no_gnu_debuglink_checksum command-line option 5
- no_ipv6_support command-line option 5
- no_kcc_classes command-line option 5
- no_message_queue command-line option 6
- no_mqd command-line option 6
- no_nptl_threads command-line option 6
- no_parallel command-line option 7
- no_team command-line option 9
- no_teamplus command-line option 9
- no_user_threads command-line option 10
- nodes_allowed command-line option 13

- nptl_threads command-line option 6

O

- option 13
- options
 - aix_use_fast_trap 2
 - tvdsrv
 - callback 11
 - serial 11
 - server 11
 - set_pw 12
 - user_threads 10

P

- parallel command-line option 7
- Parallel Virtual Machine 3
- passwords 14
 - checking 14
 - generated by tvdsrv 12
- patch_area_base command-line option 7
- patch_area_length command-line option 7
- PATH environment variable
 - for tvdsrv 11
- pid command-line option 7
- port 4142 13
- port command-line option 13
- port number 13
 - for tvdsrv 12
 - replacement 16
 - searching 13
- ports on host 12
- PVM 13
- pvm command-line option 7, 13

R

- r command-line option 7
- remote command-line option 7
- remote debugging, tvdsrv command syntax 11
- remsh command 15
- replacement characters 15

S

- s command-line option 7
- search_path command-line option 8
- search_port command-line option 13
- serial command-line option 7, 11, 13
- serial line connection 13
- server command-line option 11, 13
- server launch command 15
- servers, number of 16
- set_pw command-line option 12, 14
- set_pws command-line option 14
- shm command-line option 8, 9
- signal_handling_mode command-line option 8
- signals, handling in TotalView 8
- SLURM, control_c_quick_shutdown variable 2

stderr command-line option 9
stderr_append command-line option 9
stderr_is_stdout command-line option 9
stdin command-line option 9
stdout command-line option 9
stdout_append command-line option 9

T

team command-line option 9
teamplus command-line option 9
timeplus command-line option 9
totalview command 1
 options 1
 synopsis 1
 syntax and use 1
totalview_jobid variable 17
tvdsrvr command 11, 12, 15
 description 11
 options 12
 password 12
 PATH environment variable 11
 synopsis 11
 use with DPVM applications 13
 use with PVM applications 13
tvdsrvr.conf 13
TVDSVRLAUNCHCMD environment variable 15
tvhome command-line option 9

U

user_threads command-line option 10

V

verbosity command-line option 10, 14
verbosity setting replacement character 17
vfork()
 calling 2

W

working_directory command-line option 14

X

xterm_name command-line option 10

