

strigger(1)

Slurm Commands

strigger(1)

**NAME**

strigger – Used set, get or clear Slurm trigger information.

**SYNOPSIS**

```
strigger --set [OPTIONS...]
strigger --get [OPTIONS...]
strigger --clear [OPTIONS...]
```

**DESCRIPTION**

**strigger** is used to set, get or clear Slurm trigger information. Triggers include events such as a node failing, a job reaching its time limit or a job terminating. These events can cause actions such as the execution of an arbitrary script. Typical uses include notifying system administrators of node failures and gracefully terminating a job when it's time limit is approaching. A hostlist expression for the nodelist or job ID is passed as an argument to the program.

Trigger events are not processed instantly, but a check is performed for trigger events on a periodic basis (currently every 15 seconds). Any trigger events which occur within that interval will be compared against the trigger programs set at the end of the time interval. The trigger program will be executed once for any event occurring in that interval. The record of those events (e.g. nodes which went DOWN in the previous 15 seconds) will then be cleared. The trigger program must set a new trigger before the end of the next interval to ensure that no trigger events are missed OR the trigger must be created with an argument of "`--flags=PERM`". If desired, multiple trigger programs can be set for the same event.

**IMPORTANT NOTE:** This command can only set triggers if run by the user *SlurmUser* unless *SlurmUser* is configured as user root. This is required for the *slurmctld* daemon to set the appropriate user and group IDs for the executed program. Also note that the trigger program is executed on the same node that the *slurmctld* daemon uses rather than some allocated compute node. To check the value of *SlurmUser*, run the command:

```
scontrol show config | grep SlurmUser
```

**ARGUMENTS**

- a, --primary\_slurmctld\_failure**  
Trigger an event when the primary slurmctld fails.
- A, --primary\_slurmctld\_resumed\_operation**  
Trigger an event when the primary slurmctld resuming operation after failure.
- b, --primary\_slurmctld\_resumed\_control**  
Trigger an event when primary slurmctld resumes control.
- B, --backup\_slurmctld\_failure**  
Trigger an event when the backup slurmctld fails.
- c, --backup\_slurmctld\_resumed\_operation**  
Trigger an event when the backup slurmctld resumes operation after failure.
- C, --backup\_slurmctld\_assumed\_control**  
Trigger event when backup slurmctld assumes control.
- burst\_buffer**  
Trigger event when burst buffer error occurs.



**--clear**

Clear or delete a previously defined event trigger. The **--id**, **--jobid** or **--user** option must be specified to identify the trigger(s) to be cleared. Only user root or the trigger's creator can delete a trigger.

**-d, --down**

Trigger an event if the specified node goes into a DOWN state.

**-D, --drained**

Trigger an event if the specified node goes into a DRAINED state.

**-e, --primary\_slurmctld\_acct\_buffer\_full**

Trigger an event when primary slurmctld accounting buffer is full.

**-F, --fail**

Trigger an event if the specified node goes into a FAILING state.

**-f, --fini**

Trigger an event when the specified job completes execution.

**--flags=type**

Associate flags with the reservation. Multiple flags should be comma separated. Valid flags include:

PERM Make the trigger permanent. Do not purge it after the event occurs.

**--front\_end**

Trigger events based upon changes in state of front end nodes rather than compute nodes. Applies to Cray ALPS architectures only, where the slurmd daemon executes on front end nodes rather than the compute nodes. Use this option with either the **--up** or **--down** option.

**-g, --primary\_slurmdbd\_failure**

Trigger an event when the primary slurmdbd fails. The trigger is launched by slurmctld in the occasions it tries to connect to slurmdbd, but receives no response on the socket.

**-G, --primary\_slurmdbd\_resumed\_operation**

Trigger an event when the primary slurmdbd resumes operation after failure. This event is triggered when opening the connection from slurmctld to slurmdbd results in a response. It can happen also in different situations, periodically every 15 seconds when checking the connection status, when saving state, when agent queue is filling, and so on.

**--get** Show registered event triggers. Options can be used for filtering purposes.

**-h, --primary\_database\_failure**

Trigger an event when the primary database fails. This event is triggered when the accounting plugin tries to open a connection with mysql and it fails and the slurmctld needs the database for some operations.

**-H, --primary\_database\_resumed\_operation**

Trigger an event when the primary database resumes operation after failure. It happens when the connection to mysql from the accounting plugin is restored.



- i, --id=*id***  
Trigger ID number.
- I, --idle**  
Trigger an event if the specified node remains in an IDLE state for at least the time period specified by the **--offset** option. This can be useful to hibernate a node that remains idle, thus reducing power consumption.
- j, --jobid=*id***  
Job ID of interest. **NOTE:** The **--jobid** option can not be used in conjunction with the **--node** option. When the **--jobid** option is used in conjunction with the **--up** or **--down** option, all nodes allocated to that job will be considered the nodes used as a trigger event.
- M, --clusters=<*string*>**  
Clusters to issue commands to. Note that the SlurmDBD must be up for this option to work properly.
- n, --node[=*host*]**  
Host name(s) of interest. By default, all nodes associated with the job (if **--jobid** is specified) or on the system are considered for event triggers. **NOTE:** The **--node** option can not be used in conjunction with the **--jobid** option. When the **--jobid** option is used in conjunction with the **--up**, **--down** or **--drained** option, all nodes allocated to that job will be considered the nodes used as a trigger event. Since this option's argument is optional, for proper parsing the single letter option must be followed immediately with the value and not include a space between them. For example **"-ntux"** and not **"-n tux"**.
- N, --noheader**  
Do not print the header when displaying a list of triggers.
- o, --offset=*seconds***  
The specified action should follow the event by this time interval. Specify a negative value if action should precede the event. The default value is zero if no **--offset** option is specified. The resolution of this time is about 20 seconds, so to execute a script not less than five minutes prior to a job reaching its time limit, specify **--offset=320** (5 minutes plus 20 seconds).
- p, --program=*path***  
Execute the program at the specified fully qualified pathname when the event occurs. You may quote the path and include extra program arguments if desired. The program will be executed as the user who sets the trigger. If the program fails to terminate within 5 minutes, it will be killed along with any spawned processes.
- Q, --quiet**  
Do not report non-fatal errors. This can be useful to clear triggers which may have already been purged.
- r, --reconfig**  
Trigger an event when the system configuration changes. This is triggered when the slurmctld daemon reads its configuration file or when a node state changes.
- set** Register an event trigger based upon the supplied options. **NOTE:** An event is only triggered once. A new event trigger must be set established for future events of the same type to be processed. Triggers can only be set if the command is run by the user *SlurmUser* unless *SlurmUser* is configured as user root.



strigger(1)

Slurm Commands

strigger(1)

**-t, --time**

Trigger an event when the specified job's time limit is reached. This must be used in conjunction with the **--jobid** option.

**-u, --up**

Trigger an event if the specified node is returned to service from a DOWN state.

**--user=*user\_name\_or\_id***

Clear or get triggers created by the specified user. For example, a trigger created by user *root* for a job created by user *adam* could be cleared with an option **--user=root**. Specify either a user name or user ID.

**-v, --verbose**

Print detailed event logging. This includes time-stamps on data structures, record counts, etc.

**-V, --version**

Print version information and exit.

**OUTPUT FIELD DESCRIPTIONS****TRIG\_ID**

Trigger ID number.

**RES\_TYPE**

Resource type: *job* or *node*

**RES\_ID**

Resource ID: job ID or host names or "\*" for any host

**TYPE** Trigger type: *time* or *fini* (for jobs only), *down* or *up* (for jobs or nodes), or *drained*, *idle* or *reconfig* (for nodes only)

**OFFSET**

Time offset in seconds. Negative numbers indicated the action should occur before the event (if possible)

**USER** Name of the user requesting the action

**PROGRAM**

Pathname of the program to execute when the event occurs

**ENVIRONMENT VARIABLES**

Some **strigger** options may be set via environment variables. These environment variables, along with their corresponding options, are listed below. (Note: commandline options will always override these settings)

**SLURM\_CONF** The location of the Slurm configuration file.

**EXAMPLES**

Execute the program `/usr/sbin/primary_slurmctld_failure` whenever the primary slurmctld fails.

```
> cat /usr/sbin/primary_slurmctld_failure
#!/bin/bash
# Submit trigger for next primary slurmctld failure event
strigger --set --primary_slurmctld_failure \
```



strigger(1)

Slurm Commands

strigger(1)

```

--program=/usr/sbin/primary_slurmctld_failure
# Notify the administrator of the failure using by e-mail
/usr/bin/mail slurm_admin AT site DOT com -s Primary_SLURMCTLD_FAILURE

```

```

> strigger --set --primary_slurmctld_failure \
    --program=/usr/sbin/primary_slurmctld_failure

```

Execute the program "/usr/sbin/slurm\_admin\_notify" whenever any node in the cluster goes down. The subject line will include the node names which have entered the down state (passed as an argument to the script by Slurm).

```

> cat /usr/sbin/slurm_admin_notify
#!/bin/bash
# Submit trigger for next event
strigger --set --node --down \
    --program=/usr/sbin/slurm_admin_notify
# Notify administrator using by e-mail
/usr/bin/mail slurm_admin AT site DOT com -s NodesDown:$*

> strigger --set --node --down \
    --program=/usr/sbin/slurm_admin_notify

```

Execute the program "/usr/sbin/slurm\_suspend\_node" whenever any node in the cluster remains in the idle state for at least 600 seconds.

```

> strigger --set --node --idle --offset=600 \
    --program=/usr/sbin/slurm_suspend_node

```

Execute the program "/home/joe/clean\_up" when job 1234 is within 10 minutes of reaching its time limit.

```

> strigger --set --jobid=1234 --time --offset=-600 \
    --program=/home/joe/clean_up

```

Execute the program "/home/joe/node\_died" when any node allocated to job 1234 enters the DOWN state.

```

> strigger --set --jobid=1234 --down \
    --program=/home/joe/node_died

```

Show all triggers associated with job 1235.

```

> strigger --get --jobid=1235
TRIG_ID RES_TYPE RES_ID TYPE OFFSET USER PROGRAM
123   job  1235 time  -600  joe /home/bob/clean_up
125   job  1235 down    0  joe /home/bob/node_died

```

Delete event trigger 125.

```

> strigger --clear --id=125

```

Execute /home/joe/job\_fini upon completion of job 1237.

```

> strigger --set --jobid=1237 --fini --program=/home/joe/job_fini

```



strigger(1)

Slurm Commands

strigger(1)

**COPYING**

Copyright (C) 2007 The Regents of the University of California. Produced at Lawrence Livermore National Laboratory (cf, **DISCLAIMER**).

Copyright (C) 2008–2010 Lawrence Livermore National Security.

Copyright (C) 2010–2013 SchedMD LLC.

This file is part of Slurm, a resource management program. For details, see <https://slurm.schedmd.com/>.

Slurm is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Slurm is distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of **MERCHANTABILITY** or **FITNESS FOR A PARTICULAR PURPOSE**. See the GNU General Public License for more details.

**SEE ALSO**

**scontrol**(1), **sinfo**(1), **squeue**(1)

