## NAME

slurm_kill_job, slurm_kill_job_step,
slurm_signal_job, slurm_signal_job_step,
slurm_terminate_job_step − Slurm job signal calls

## SYNTAX

#include <slurm/slurm.h>

int **slurm_kill_job** (
uint32_t *job_id*,
uint16_t *signal*,
uint16_t *batch_flag*
);

int **slurm_kill_job_step** (
uint32_t *job_id*,
uint32_t *job_step_id*,
uint16_t *signal*
);

int **slurm_signal_job** (
uint32_t *job_id*,
uint16_t *signal*
);

int **slurm_signal_job_step** (
uint32_t *job_id*,
uint32_t *job_step_id*,
uint16_t *signal*
);

int **slurm_terminate_job_step** (
uint32_t *job_id*,
uint32_t *job_step_id*,
);

## ARGUMENTS

*batch_flag* If non−zero then signal only the batch job shell.

*job_id* Slurm job id number.

*job_step_id*
Slurm job step id number.

*signal* Signal to be sent to the job or job step.

## DESCRIPTION

**slurm_kill_job** Request that a signal be sent to either the batch job shell (if *batch_flag* is non−zero) or all steps of the specified job. If the job is pending and the signal is SIGKILL, the job will be terminated immediately. This function may only be successfully executed by the job's owner or user root.

**slurm_kill_job_step** Request that a signal be sent to a specific job step. This function may only be successfully executed by the job's owner or user root.

**slurm_signal_job** Request that the specified signal be sent to all steps of an existing job.

**slurm_signal_job_step** Request that the specified signal be sent to an existing job step.

**slurm_terminate_job_step** Request that terminates a job step by sending a REQUEST_TERMINATE_TASKS rpc to all slurmd of a job step.

## RETURN VALUE

On success, zero is returned. On error, −1 is returned, and Slurm error code is set appropriately.

## ERRORS

**SLURM_PROTOCOL_VERSION_ERROR** Protocol version has changed, re−link your code.

**ESLURM_DEFAULT_PARTITION_NOT_SET** the system lacks a valid default partition.

**ESLURM_INVALID_JOB_ID** the requested job id does not exist.



**ESLURM_JOB_SCRIPT_MISSING** the *batch_flag* was set for a non−batch job.

**ESLURM_ALREADY_DONE** the specified job has already completed and can not be modified.

**ESLURM_ACCESS_DENIED** the requesting user lacks authorization for the requested action (e.g. trying to delete or modify another user's job).

**ESLURM_INTERCONNECT_FAILURE** failed to configure the node interconnect.

**SLURM_PROTOCOL_SOCKET_IMPL_TIMEOUT** Timeout in communicating with Slurm controller.

## NOTE

These functions are included in the libslurm library, which must be linked to your process for use (e.g. "cc −lslurm myprog.c").

## COPYING

Copyright (C) 2002 The Regents of the University of California. Produced at Lawrence Livermore National Laboratory (cf, DISCLAIMER). CODE−OCEC−09−009. All rights reserved.

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

**scancel**(1), **slurm_get_errno**(3), **slurm_perror**(3), **slurm_strerror**(3)

