NAME

slurm_free_job_step_info_response_msg, slurm_get_job_steps, slurm_print_job_step_info, slurm_print_job_step_info_msg − Slurm job step information reporting functions

SYNTAX

#include <stdio.h>
#include <slurm/slurm.h>

void **slurm_free_job_step_info_response_msg** (
job_step_info_response_msg_t **job_step_info_msg_ptr*
);

void **slurm_get_job_steps** (
time_t **update_time*,
uint32_t *job_id*,
uint32_t *step_id*,
job_step_info_response_msg_t ***job_step_info_msg_pptr*,
uint16_t *show_flags*
);

void **slurm_print_job_step_info** (
FILE **out_file*,
*job_step_info_t *job_step_ptr,*
int one_liner
);

void **slurm_print_job_step_info_msg** (
FILE **out_file*,
*job_step_info_response_msg_t *job_step_info_msg_ptr,*
int one_liner
);

ARGUMENTS

job_id Specifies a slurm job ID. A value of zero implies all jobs.

job_step_info_msg_pptr
Specifies the double pointer to the structure to be created and filled with the time of the last node update, a record count, and detailed information about each job step specified. Detailed job step information is written to fixed sized records and includes: job_id, step_id, node names, etc. See slurm.h for full details on the data structure's contents.

job_step_info_msg_ptr
Specifies the pointer to the structure created by the function **slurm_get_job_steps**.

job_step_ptr
Specifies a pointer to a single job step records from the *job_step_info_msg_pptr* data structure.

one_liner
Print one record per line if non−zero.

out_file Specifies the file to print data to.

show_flags
Job filtering flags, may be ORed. Information about job steps in partitions that are configured as hidden and partitions that the user's group is unable to utilize are not reported by default. The **SHOW_ALL** flag will cause information about job steps in all partitions to be displayed.

step_id Specifies a slurm job step ID. A value of zero implies all job steps.

update_time
For all of the following informational calls, if update_time is equal to or greater than the last time changes where made to that information, new information is not returned. Otherwise all the configuration. job, node, or partition records are returned.



DESCRIPTION

slurm_free_job_step_info_response_msg Release the storage generated by the **slurm_get_job_steps** function.

slurm_get_job_steps Loads into details about job steps that satisfy the *job_id* and/or *step_id* specifications provided if the data has been updated since the *update_time* specified.

slurm_print_job_step_info Prints the contents of the data structure describing a single job step records from the data loaded by the lurm_get_job_steps function.

slurm_print_job_step_info_msg Prints the contents of the data structure describing all job step records loaded by the lurm_get_job_steps function.

RETURN VALUE

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

ERRORS

SLURM_NO_CHANGE_IN_DATA Data has not changed since **update_time**.

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

SLURM_PROTOCOL_SOCKET_IMPL_TIMEOUT Timeout in communicating with Slurm controller.

EXAMPLE

```
#include <stdio.h>
#include <stdlib.h>
#include <slurm/slurm.h>
#include <slurm/slurm_errno.h>

int main (int argc, char *argv[])
{
        int i;
        job_step_info_response_msg_t * step_info_ptr = NULL;
        job_step_info_t * step_ptr;

        /* get and dump some job information */
        if ( slurm_get_job_steps ((time_t) NULL, 0, 0,
                        &step_info_ptr, SHOW_ALL) ) {
                slurm_perror ("slurm_get_job_steps error");
                exit (1);
        }

        /* The easy way to print... */
        slurm_print_job_step_info_msg (stdout,
                                step_info_ptr, 0);

        /* A harder way.. */
        for (i = 0; i < step_info_ptr−>job_step_count; i++) {
                step_ptr = &step_info_ptr−>job_steps[i];
                slurm_print_job_step_info(stdout, step_ptr, 0);
        }

        /* The hardest way. */
        printf ("Steps updated at %lx, record count %d\n",
              step_info_ptr−>last_update,
              step_info_ptr−>job_step_count);
        for (i = 0; i < step_info_ptr−>job_step_count; i++) {
                printf ("JobId=%u StepId=%u\n",
                                step_info_ptr−>job_steps[i].job_id,
                                step_info_ptr−>job_steps[i].step_id);
        }

        slurm_free_job_step_info_response_msg(step_info_ptr);
        exit (0);
}
```

NOTES

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

Some data structures contain index values to cross−reference each other. If the *show_flags* argument is not set to SHOW_ALL when getting this data, these index values will be invalid.

The **slurm_hostlist_** functions can be used to convert Slurm node list expressions into a collection of individual node names.

COPYING

Copyright (C) 2002−2006 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

scontrol(1), **squeue**(1), **slurm_hostlist_create**(3), **slurm_hostlist_shift**(3), **slurm_hostlist_destroy**(3), **slurm_get_errno**(3), **slurm_load_jobs**(3), **slurm_perror**(3), **slurm_strerror**(3)

