NAME

slurm_free_partition_info_msg, slurm_load_partitions, slurm_print_partition_info, slurm_print_partition_info_msg − Slurm partition information reporting functions

SYNTAX

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

void **slurm_free_partition_info_msg** (
partition_info_msg_t **partition_info_msg_ptr*
);

int **slurm_load_partitions** (
time_t *update_time*,
partition_info_msg_t ***partition_info_msg_pptr*,
uint16_t *show_flags*
);

void **slurm_print_partition_info** (
FILE **out_file*,
partition_info_t **partition_ptr*,
int *one_liner*
);

void **slurm_print_partition_info_msg** (
FILE **out_file*,
partition_info_msg_t **partition_info_msg_ptr*,
int *one_liner*
);

ARGUMENTS

one_liner
Print one record per line if non−zero.

out_file Specifies the file to print data to.

partition_info_msg_pptr
Specifies the double pointer to the structure to be created and filled with the time of the last partition update, a record count, and detailed information about each partition. Detailed partition information is written to fixed sized records and includes: name, state, job time limit, job size limit, node names, indexes into the node table, etc. In the case of indexes into the node table, this is an array of integers with pairs of start and end index number into the node information records and the data is terminated with a value of −1. See slurm.h for full details on the data structure's contents.

partition_info_msg_ptr
Specifies the pointer to the structure created by **slurm_load_partitions**.

show_flags
Job filtering flags, may be ORed. Information about 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 partitions to be displayed. Only information about partitions on the local cluster will be returned unless the cluster is in a federation and the **SHOW_ALL** flag is set.

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_partition_info_msg Release the storage generated by the **slurm_load_partitions** function.

slurm_load_partitions Returns a partition_info_msg_t that contains an update time, record count, and



array of partition_table records for all partitions.

slurm_print_partition_info Prints the contents of the data structure describing a single partition records from the data loaded by the **slurm_load_partitions** function.

slurm_print_partition_info_msg Prints the contents of the data structure describing all partition records loaded by the **slurm_load_partitions** 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;
        partition_info_msg_t *part_info_ptr = NULL;
        partition_info_t *part_ptr;

        /* get and dump some partition information */
        if (slurm_load_partitions((time_t)NULL,
                        &part_info_ptr, SHOW_ALL)) {
                slurm_perror ("slurm_load_partitions error");
                exit (1);
        }

        /* The easy way to print... */
        slurm_print_partition_info_msg (stdout,
                                part_info_ptr, 0);

        /* A harder way.. */
        for (i = 0; i < part_info_ptr->record_count; i++) {
                part_ptr = &part_info_ptr->partition_array[i];
                slurm_print_partition_info(stdout, part_ptr, 0);
        }

        /* The hardest way. */
        printf("Partitions updated at %lx, records=%d\n",
             part_info_ptr->last_update,
             part_info_ptr->record_count);
        for (i = 0; i < part_info_ptr->record_count; i++) {
                printf ("PartitionName=%s Nodes=%s\n",
                        part_info_ptr->partition_array[i].name,
                        part_info_ptr->partition_array[i].nodes );
        }

        slurm_free_partition_info_msg (part_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), **sinfo**(1), **squeue**(1), **slurm_hostlist_create**(3), **slurm_hostlist_shift**(3), **slurm_hostlist_destroy**(3), **slurm_get_errno**(3), **slurm_load_node**(3), **slurm_perror**(3), **slurm_strerror**(3)

