

**NAME**

`slurm_load_reservations`, `slurm_free_reservation_info_msg`, `slurm_print_reservation_info`,  
`slurm_sprint_reservation_info`, `slurm_print_reservation_info_msg` – Slurm reservation information  
reporting functions

**SYNTAX**

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

int slurm_load_reservations (
    time_t update_time,
    reserve_info_msg_t **reservation_info_msg_ptr
);

void slurm_free_reservation_info_msg (
    reserve_info_msg_t *reservation_info_msg_ptr
);

void slurm_print_reservation_info (
    FILE *out_file,
    reserve_info_t *reservation_ptr,
    int one_liner
);

char * slurm_sprint_reservation_info (
    reserve_info_t *reservation_ptr,
    int one_liner
);

void slurm_print_reservation_info_msg (
    FILE *out_file,
    reserve_info_msg_t *reservation_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.

*reservation\_info\_msg\_ptr*

Specifies the double pointer to the structure to be created and filled with the time of the last reservation update, a record count, and detailed information about each reservation. Detailed reservation information is written to fixed sized records and includes: reservation name, time limits, access restrictions, etc. See `slurm.h` for full details on the data structure's contents.

*reservation\_info\_msg\_ptr*

Specifies the pointer to the structure created by **`slurm_load_reservations`**.

*update\_time*

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

**DESCRIPTION**

**`slurm_load_reservations`** Returns a `reserve_info_msg_t` that contains an update time, record count, and array of `reservation_table` records for all reservations.

**`slurm_free_reservation_info_msg`** Release the storage generated by the **`slurm_load_reservations`** function.

**`slurm_print_reservation_info`** Prints the contents of the data structure describing one of the reservation records from the data loaded by the **`slurm_load_reservations`** function.

**`slurm_sprint_reservation_info`** Prints the same info as **`slurm_print_reservation_info`**, but prints to a string that must be freed by the caller, rather than printing to a file.



**slurm\_print\_reservation\_info\_msg** Prints the contents of the data structure describing all reservation records loaded by the **slurm\_load\_reservations** 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;
    reserve_info_msg_t *res_info_ptr = NULL;
    reserve_info_t *res_ptr;

    /* get and dump all reservation information */
    if (slurm_load_reservations((time_t)NULL,
                               &res_info_ptr)) {
        slurm_perror ("slurm_load_reservations error");
        exit (1);
    }

    /* The easy way to print... */
    slurm_print_reservation_info_msg(stdout,
                                     res_info_ptr, 0);

    /* A harder way.. */
    for (i = 0; i < res_info_ptr->record_count; i++) {
        res_ptr = &res_info_ptr->reservation_array[i];
        slurm_print_reservation_info(stdout, res_ptr, 0);
    }

    /* The hardest way. */
    printf("reservations updated at %lx, records=%d\n",
           res_info_ptr->last_update,
           res_info_ptr->record_count);
    for (i = 0; i < res_info_ptr->record_count; i++) {
        printf ("reservationName=%s Nodes=%s\n",
                res_info_ptr->reservation_array[i].name,
                res_info_ptr->reservation_array[i].node_list );
    }

    slurm_free_reservation_info_msg (res_info_ptr);
    return 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").

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)`

