Grid Engine Testsuite

Documentation taken from source code
Edition for Version
11 June 2001

1 check

1.1 add_proc_error

NAME

```
add_proc_error -- append testsuite error message
```

SYNOPSIS

```
add_proc_error { proc_name result text }
```

FUNCTION

```
This procedure adds a new error to the global error arrays for the global
procedures.

So a test programmer doesn't have to set the error states after calling
a global procedure which uses add_proc_error. Each global procedure
set the error state by itself.

The test run will report ALL global errors and doesn't set the test run
to a correct state if such an error is reported.

Some global procedures have an optional flag to switch off the global
error report. For some cases it is necessary to turn off the error
reporting. (e.g. forced timeout test)
```

INPUTS

```
proc_name - name of the calling procedure
result    - error state (e.g. -1)
text      - error text (e.g. "open file xxx failed)
```

RESULT

```
no result
```

SEE ALSO

See Section 1.58 [check set_error], page 30.

1.2 ask_user_yes_or_no

NAME

```
ask_user_yes_or_no -- ???
```

SYNOPSIS

```
ask_user_yes_or_no { question }
```

FUNCTION

```
???
```

INPUTS

```
question - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

1.3 auto_reschedule_cleanup

NAME

```
auto_reschedule_cleanup -- ???
```

SYNOPSIS

```
auto_reschedule_cleanup { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

1.4 auto_reschedule_setup

NAME

```
auto_reschedule_setup -- ???
```

SYNOPSIS

```
auto_reschedule_setup { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

1.5 auto_reschedule_unknown_check

NAME

```
auto_reschedule_unknown_check -- ???
```

SYNOPSIS

```
auto_reschedule_unknown_check { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

1.6 auto_reschedule_unknown_check_master

NAME

```
auto_reschedule_unknown_check_master -- ???
```

SYNOPSIS

```
auto_reschedule_unknown_check_master { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.7 calc_space

NAME

```
calc_space -- ???
```

SYNOPSIS

```
calc_space { space name }
```

FUNCTION

```
???
```

INPUTS

```
space - ???
name  - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.8 change_dir

NAME

```
change_dir -- ???
```

SYNOPSIS

```
change_dir { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.9 check_root_access

NAME

```
check_root_access -- ???
```

SYNOPSIS

```
check_root_access { path }
```

FUNCTION

```
???
```

INPUTS

```
path - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.10 clean_up_globals

NAME

```
clean_up_globals -- ???
```

SYNOPSIS

```
clean_up_globals { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.11 clear_screen

NAME

```
clear_screen -- ???
```

SYNOPSIS

```
clear_screen { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.12 cluster_perf_make_analysis

NAME

```
cluster_perf_make_analysis() -- ???
```

SYNOPSIS

```
cluster_perf_make_analysis { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.13 compile_source

NAME

```
compile_source() -- ???
```

SYNOPSIS

```
compile_source { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

1.14 create_error_message

NAME

```
create_error_message -- ???
```

SYNOPSIS

```
create_error_message { error_array }
```

FUNCTION

```
???
```

INPUTS

```
error_array - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

1.15 create_report

NAME

```
create_report -- ???
```

SYNOPSIS

```
create_report { file goodbad }
```

FUNCTION

```
???
```

INPUTS

```
file    - ???
goodbad - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.16 debug_puts

NAME

```
debug_puts -- ???
```

SYNOPSIS

```
debug_puts { args }
```

FUNCTION

```
???
```

INPUTS

```
args - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.17 delete_result

NAME

```
delete_result -- ???
```

SYNOPSIS

```
delete_result { path runtime level }
```

FUNCTION

```
???
```

INPUTS

```
path    - ???
runtime - ???
level   - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.18 delete_tests

NAME

```
delete_tests -- ???
```

SYNOPSIS

```
delete_tests { path { only_if_not_there 0 } }
```

FUNCTION

```
???
```

INPUTS

```
path                    - ???
{ only_if_not_there 0 } - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.19 do_wait

NAME

```
do_wait -- ???
```

SYNOPSIS

```
do_wait { time }
```

FUNCTION

```
???
```

INPUTS

```
time - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.20 edit_defaults

NAME

```
edit_defaults -- ???
```

SYNOPSIS

```
edit_defaults { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.21 format_output

NAME

```
format_output -- ???
```

SYNOPSIS

```
format_output { prefix size text }
```

FUNCTION

```
???
```

INPUTS

```
prefix - ???
size   - ???
text   - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.22 get_check_dirs

NAME

```
get_check_dirs -- ???
```

SYNOPSIS

```
get_check_dirs { path }
```

FUNCTION

```
???
```

INPUTS

```
path - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.23 get_check_name

NAME

```
get_check_name -- ???
```

SYNOPSIS

```
get_check_name { path }
```

FUNCTION

```
???
```

INPUTS

```
path - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.24 get_current_working_dir

NAME

```
get_current_working_dir -- ???
```

SYNOPSIS

```
get_current_working_dir { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.25 get_max_level_count

NAME

```
get_max_level_count -- ???
```

SYNOPSIS

```
get_max_level_count { path }
```

FUNCTION

```
???
```

INPUTS

```
path - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.26 get_root_passwd

NAME

```
get_root_passwd -- return root password
```

SYNOPSIS

```
get_root_passwd { }
```

FUNCTION

```
This procedure returns the root password, typed in by the user.
```

RESULT

```
string with root password
```

SEE ALSO

See Section 1.30 [check have_root_passwd], page 15.
See Section 1.59 [check set_root_passwd], page 30.

1.27 get_run_level_name

NAME

```
get_run_level_name -- ???
```

SYNOPSIS

```
get_run_level_name { level }
```

FUNCTION

```
???
```

INPUTS

```
level - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.28 get_test_result

NAME

```
get_test_result -- ???
```

SYNOPSIS

```
get_test_result { filename }
```

FUNCTION

```
???
```

INPUTS

```
filename - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.29 get_user_input

NAME

```
get_user_input -- ???
```

SYNOPSIS

```
get_user_input { what }
```

FUNCTION

```
???
```

INPUTS

```
what - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.30 have_root_passwd

NAME

```
have_root_passwd -- is root password available ?
```

SYNOPSIS

```
have_root_passwd { }
```

FUNCTION

```
test if root password was typed in
```

INPUTS

```
0  : root password should be ok
-1 : no root access
```

SEE ALSO

See Section 1.59 [check set_root_passwd], page 30.

See Section 1.26 [check get_root_passwd], page 13.

1.31 have_ssh_access

NAME

```
have_ssh_access -- is ssh accessable ?
```

SYNOPSIS

```
have_ssh_access { }
```

FUNCTION

```
This procedure tries to get a ssh (secure shell) connection to each execd
host from the cluster. The result of this test is stored in a global
variable so the next call will not cause the connection test again.
```

RESULT

```
0: no ssh access
1: ok
```

SEE ALSO

See '/'

1.32 init_level

NAME

```
init_level -- ???
```

SYNOPSIS

```
init_level { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

NAME

```
init_level -- ???
```

SYNOPSIS

```
init_level { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.33 is_level_enabled

NAME

```
is_level_enabled -- ???
```

SYNOPSIS

```
is_level_enabled { level_nr }
```

FUNCTION

```
???
```

INPUTS

```
level_nr - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.34 is_version_ok

NAME

```
is_version_ok() -- ???
```

SYNOPSIS

```
is_version_ok { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

NAME

```
is_version_ok() -- ???
```

SYNOPSIS

```
is_version_ok { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.35 load_defaults

NAME

```
load_defaults -- ???
```

SYNOPSIS

```
load_defaults { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.36 lock_testsuite

NAME

```
lock_testsuite -- ???
```

SYNOPSIS

```
lock_testsuite { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.37 mail_report

NAME

```
mail_report -- send mail
```

SYNOPSIS

```
mail_report { subject body }
```

FUNCTION

```
This procedure sends an e-mail to the e-mail-address configured
with the global variables CHECK_REPORT_EMAIL_CC  and
CHECK_REPORT_EMAIL_TO. Subject and body of the mail is taken
from the parameters subject and body.
```

INPUTS

```
subject - e-mail subject text
body    - e-mail body text
```

SEE ALSO

See Section 1.57 [check send_mail], page 29.

1.38 menu

NAME

```
menu -- ???
```

SYNOPSIS

```
menu { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.39 print_menu_header

NAME

```
print_menu_header -- ???
```

SYNOPSIS

```
print_menu_header { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.40 print_results

NAME

```
print_results -- ???
```

SYNOPSIS

```
print_results { ckpath where }
```

FUNCTION

```
???
```

INPUTS

```
ckpath - ???
where  - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.41 read_edit_defaults_file

NAME

```
read_edit_defaults_file -- ???
```

SYNOPSIS

```
read_edit_defaults_file { filename }
```

FUNCTION

```
???
```

INPUTS

```
filename - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.42 reschedule_checkpointing

NAME

```
reschedule_checkpointing -- ???
```

SYNOPSIS

```
reschedule_checkpointing { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

1.43 reschedule_cleanup

NAME

```
reschedule_cleanup -- ???
```

SYNOPSIS

```
reschedule_cleanup { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

1.44 reschedule_deleted_job

NAME

```
reschedule_deleted_job -- ???
```

SYNOPSIS

```
reschedule_deleted_job { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.45 reschedule_pe_jobs

NAME

```
reschedule_pe_jobs -- ???
```

SYNOPSIS

```
reschedule_pe_jobs { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.46 reschedule_qsh_qlogin_qrsh_qrlogin

NAME

```
reschedule_qsh_qlogin_qrsh_qrlogin -- ???
```

SYNOPSIS

```
reschedule_qsh_qlogin_qrsh_qrlogin { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.47 reschedule_setup

NAME

```
reschedule_setup -- ???
```

SYNOPSIS

```
reschedule_setup { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.48 reschedule_submit_jobs

NAME

```
reschedule_submit_jobs -- ???
```

SYNOPSIS

```
reschedule_submit_jobs { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

1.49 run_all_continuously

NAME

```
run_all_continuously -- ???
```

SYNOPSIS

```
run_all_continuously { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

1.50 run_test

NAME

```
run_test -- ???
```

SYNOPSIS

```
run_test { path runcompleted {run_single_test "all"} }
```

FUNCTION

```
???
```

INPUTS

```
path                     - ???
runcompleted             - ???
{run_single_test "all"} - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.51 run_test_level

NAME

```
run_test_level -- ???
```

SYNOPSIS

```
run_test_level { path runcompleted level {do_save 1} }
```

FUNCTION

```
???
```

INPUTS

```
path         - ???
runcompleted - ???
level        - ???
{do_save 1}  - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.52 run_tests

NAME

```
run_tests -- ???
```

SYNOPSIS

```
run_tests { path runcompleted }
```

FUNCTION

```
???
```

INPUTS

```
path         - ???
runcompleted - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.53 save_defaults

NAME

```
save_defaults -- ???
```

SYNOPSIS

```
save_defaults { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.54 save_result

NAME

```
save_result -- ???
```

SYNOPSIS

```
save_result { path runtime level }
```

FUNCTION

```
???
```

INPUTS

```
path    - ???
runtime - ???
level   - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.55 scheduler_perf_make_analysis

NAME

```
scheduler_perf_make_analysis() -- ???
```

SYNOPSIS

```
scheduler_perf_make_analysis { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.56 select_runlevel

NAME

```
select_runlevel -- ???
```

SYNOPSIS

```
select_runlevel { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.57 send_mail

NAME

```
send_mail -- send mail
```

SYNOPSIS

```
send_mail { address cc subject body }
```

FUNCTION

```
This procedure calls the mailx binary by using remote shell to
send an e-mail.
```

INPUTS

```
address - e-mail address
cc      - e-mail CC address
subject - e-mail subject text
body    - e-mail body text
```

SEE ALSO

See Section 1.37 [check mail_report], page 19.

1.58 set_error

NAME

```
set_error -- set error for current check
```

SYNOPSIS

```
set_error { erno errtext }
```

FUNCTION

```
This procedure simply sets the global variables check_errno and
check_errstr to the given parameters. Beyond it the procedure
add_proc_error is called in order to append the errors to the global error
list.
```

INPUTS

```
erno    - integer
          0  = no error
          -1 = error, but the check will run till end
          -2 = error, the current check will stop (no further check functi
               is called)
          -3 = warning, (e.g. test can not run on this host)

errtext - short error description
```

EXAMPLE

```
set_error 0 "ok"  ;# Test is "OK"
```

SEE ALSO

See Section 1.1 [check add_proc_error], page 1.

1.59 set_root_passwd

NAME

```
set_root_passwd -- ask user for root password
```

SYNOPSIS

```
set_root_passwd { }
```

FUNCTION

```
This procedure reads in the root password from stdin. If the root password
is not used ( ssh access garanted) the procedure returns immediately.
The root password is tested with an id call as root on the local machine.
```

SEE ALSO

See Section 1.30 [check have_root_passwd], page 15.
See Section 1.26 [check get_root_passwd], page 13.

1.60 setup

NAME

```
setup -- ???
```

SYNOPSIS

```
setup { {do_only_hostname_resolving 0} }
```

FUNCTION

```
???
```

INPUTS

```
{do_only_hostname_resolving 0} - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.61 show_proc_error

NAME

```
show_proc_error -- ???
```

SYNOPSIS

```
show_proc_error { result new_error }
```

FUNCTION

```
???
```

INPUTS

```
result    - ???
new_error - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.62 show_test

NAME

```
show_test -- ???
```

SYNOPSIS

```
show_test { path full }
```

FUNCTION

```
???
```

INPUTS

```
path - ???
full - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.63 show_tests

NAME

```
show_tests -- ???
```

SYNOPSIS

```
show_tests { path full }
```

FUNCTION

```
???
```

INPUTS

```
path - ???
full - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.64 source_procedures

NAME

```
source_procedures -- ???
```

SYNOPSIS

```
source_procedures { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.65 unlock_testsuite

NAME

```
unlock_testsuite -- ???
```

SYNOPSIS

```
unlock_testsuite { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.66 validate_needs

NAME

```
validate_needs -- ???
```

SYNOPSIS

```
validate_needs { needs }
```

FUNCTION

```
???
```

INPUTS

```
needs - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.67 wait_for_enter

NAME

```
wait_for_enter -- ???
```

SYNOPSIS

```
wait_for_enter { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.68 wait_for_start_time

NAME

```
wait_for_start_time -- ???
```

SYNOPSIS

```
wait_for_start_time { substring }
```

FUNCTION

```
???
```

INPUTS

```
substring - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

1.69 write_edit_defaults_file

NAME

```
write_edit_defaults_file -- ???
```

SYNOPSIS

```
write_edit_defaults_file { filename { unique_file def_edit_file } }
```

FUNCTION

```
???
```

INPUTS

```
filename                          - ???
{ unique_file def_edit_file } - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

2 control_procedures

2.1 get_ps_info

NAME

```
get_ps_info -- get ps output on remote or local host
```

SYNOPSIS

```
get_ps_info { { pid 0 } { host "local"} { variable ps_info }
{additional_run 0} }
```

FUNCTION

```
This procedure will call ps on the host given and parse the output. All
information is stored in a special array. If no variable parameter is
given the array has the name ps_info
```

INPUTS

```
{ pid 0 }            - set pid for ps_info($pid,error) the
                       ps_info([given pid],error) array is always set when
                       the pid is given. You have always access to
                       ps_info($pid,error)
{ host "local"}      - host on which the ps command should be started
{ variable ps_info } - array name where the ps command output should be
                       stored the default for this value is "ps_info"
{additional_run 0}   - if it is neccessary to start more than one ps comman
                       to get the full information this number is used to b
                       able to differ the recursive subcalls. So this
                       parameter is only set when the procedure calls itsel
                       again.
```

RESULT

```
The procedure returns an 2 dimensional array with following entries:

If the parameter pid was set to 12 then ps_info(12,error) exists after
calling this procedure ps_info(12,error) is set to 0 when the pid 12 exists
otherwise it is set to -1

when ps_info(12,error) exists the following indicies are available:

ps_info(12,string)
ps_info(12,index_names)
ps_info(12,pgid)
ps_info(12,ppid)
ps_info(12,uid)
ps_info(12,state)
ps_info(12,stime)
ps_info(12,vsz)
ps_info(12,time)
ps_info(12,command)

every output of the ps command is stored into these indicies:
(I is the line number (or index) of the output)

ps_info(proc_count)   : number of processes (line count of ps command)
ps_info(pid,I)        : pid of process
ps_info(pgid,I)       : process group id
ps_info(ppid,I)       : parent pid
ps_info(uid,I)        : user id
ps_info(state,I)      : state
ps_info(stime,I)      : start time
ps_info(vsz,I)        : virtual size
ps_info(time,I)       : cpu time
ps_info(command,I)    : command arguments of process
ps_info(string,I)     : complete line
```

EXAMPLE

```
get process group id of pid 3919:

get_ps_info 3919 fangorn
if {$ps_info(3919,error) == 0} {
   puts "process group id of pid 3919 is $ps_info(3919,pgid)"
} else {
   puts "pid 3919 not found!"
}



print out all pids on local host:

get_ps_info
for {set i 0} {$i < $ps_info(proc_count) } {incr i 1} {
   puts "ps_info(pid,$i)     = $ps_info(pid,$i)"
}
```

NOTES

```
o additional_run is for glinux at this time
o additionan_run is a number from 0 up to xxx at the end of the procedure
  it will start again a ps command with other information in order to mix
  up the information into one resulting list

o this procedure should run on following platforms:
  solaris64, solaris, osf4, tru64, irix6, aix43, aix42, hp10, hp11, glinux
  and alinux
```

BUGS

```
???
```

SEE ALSO

See Section 2.3 [control_procedures ps_grep], page 38.

2.2 handle_vi_edit

NAME

```
handle_vi_edit -- sending vi commands to application
```

SYNOPSIS

```
handle_vi_edit { prog_binary prog_args vi_command_sequence
expected_result {additional_expected_result "___ABCDEFG___"}
{additional_expected_result2 "___ABCDEFG___"} }
```

FUNCTION

```
Start an application which and send special command strings to it. Wait
and parse the application output.
```

INPUTS

```
prog_binary                                   - application binary to star
                                                (e.g. qconf)
prog_args                                     - application arguments (e.g
                                                -mconf)
vi_command_sequence                           - list of vi command sequenc
                                                (e.g.
                                                {:%s/^$elem .*$/$elem 10/\
expected_result                               - program output in no error
                                                case (e.g. modified)
{additional_expected_result "___ABCDEFG___"}  - additional expected_result
{additional_expected_result2 "___ABCDEFG___"} - additional expected_result
```

RESULT

```
 0 when the output of the application contents the expected_result
-1 on timeout
-2 on additional_expected_result
-3 on additional_expected_result2
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

2.3 ps_grep

NAME

```
ps_grep -- call get_ps_info and return only expected ps information
```

SYNOPSIS

```
ps_grep { forwhat { host "local" } { variable ps_info } }
```

FUNCTION

```
This procedure will call the get_ps_info procedure. It will parse the
get_ps_info result for the given strings and return only those process
ids which match.
```

INPUTS

```
forwhat                  - search string (e.g. binary name)
{ host "local" }         - host on which the ps command should be called
{ variable ps_info } - variable name to store the result (default ps_info)
```

RESULT

```
returns a list of indexes where the search string matches the ps output.
```

EXAMPLE

```
set myprocs [ ps_grep "execd" "fangorn" ]

puts "execd's on fangorn index list: $myprocs"

foreach elem $myprocs {
  puts $ps_info(string,$elem)
}

output of example:

execd's on fangorn index list: 34 39 50 59 61
2530   140     1   259 S Sep12  1916 00:00:14 /sge_s/glinux/sge_execd
7700   142     1   339 S Sep13  2024 00:03:49 /vol2/bin/glinux/sge_execd
19159     0     1     0 S Sep14  1772 00:31:09 /vol/bin/glinux/sgeee_execd
24148     0     1     0 S Sep14  2088 00:06:23 bin/glinux/sge_execd
15085     0     1     0 S Sep14  1904 00:27:04 /vol2/glinux/sgeee_execd
```

NOTES

```
look at get_ps_info procedure for more information!
```

BUGS

```
???
```

SEE ALSO

See Section 2.1 [control_procedures get_ps_info], page 36.

3 file_procedures

3.1 cleanup_spool_dir

NAME

```
cleanup_spool_dir -- create or cleanup spool directory for master/execd
```

SYNOPSIS

```
cleanup_spool_dir { topleveldir subdir }
```

FUNCTION

```
This procedure will create or cleanup old entries in the qmaster or execd
spool directory
```

INPUTS

```
topleveldir - path to spool toplevel directory ( updir of qmaster and exec
subdir      - this paramter is master or execd
```

RESULT

```
if ok the procedure returns the correct spool directory. It returns  on
error
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 3.5 [file_procedures delete_directory], page 42.

3.2 copy_directory

NAME

```
copy_directory -- copy a directory recursively
```

SYNOPSIS

```
copy_directory { source target }
```

FUNCTION

```
This procedure will copy the given source directory to the target
directory. The content of the target dir is deleted if it exists.
(calling delete_directory, which will make a secure copy in the testsuite
trash folder).
```

INPUTS

```
source - path to the source directory
target - path to the target directory
```

RESULT

```
no results
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 3.5 [file_procedures delete_directory], page 42.

3.3 create_shell_script

NAME

```
create_shell_script -- create a /bin/sh script file
```

SYNOPSIS

```
create_shell_script { scriptfile exec_command exec_arguments }
```

FUNCTION

```
This procedure generates a script which will execute the given command.
The script will restore the testsuite and SGE environment first. It will
also echo _start_mark_:(x) and _exit_status_:(x) where x is the exit
value from the started command.
```

INPUTS

```
scriptfile     - full path and name of scriptfile to generate
exec_command   - command to execute
exec_arguments - command parameters
```

RESULT

```
no results
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 3.9 [file_procedures get_dir_names], page 44.

3.4 del_job_files

NAME

```
del_job_files -- delete files that conain a specific jobid
```

SYNOPSIS

```
del_job_files { jobid job_output_directory expected_file_count }
```

FUNCTION

```
This function reads in the job_output_directory and is looking for
filenames that contain the given jobid. If after a maximum time of 120
seconds not the number of expected_file_count is reached, a timeout will
happen. After that the files are deleted.
```

INPUTS

```
jobid                - jobid of job which has created the output file
job_output_directory - path to the directory that contains the output files
expected_file_count  - number of output files that are expected
```

RESULT

```
returns the number of deleted files
```

SEE ALSO

See Section 3.9 [file_procedures get_dir_names], page 44.

3.5 delete_directory

NAME

```
delete_directory -- move/copy directory to testsuite trashfolder
```

SYNOPSIS

```
delete_directory { path }
```

FUNCTION

```
This procedure will move/copy the given directory to the testsuite's
trashfolder (Directory testsuite_trash in the testsuite root directory).
```

INPUTS

```
path - full directory path
```

RESULT

```
-1 on error, 0 ok
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 3.5 [file_procedures delete_directory], page 42.

3.6 delete_file

NAME

```
delete_file -- move/copy file to testsuite trashfolder
```

SYNOPSIS

```
delete_file { filename }
```

FUNCTION

```
This procedure will move/copy the file to the testsuite's trashfolder
(Directory testsuite_trash in the testsuite root directory).
```

INPUTS

```
filename - full path file name of file
```

RESULT

```
no results
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 3.5 [file_procedures delete_directory], page 42.

3.7 delete_file_at_startup

NAME

```
delete_file_at_startup -- delete old temp files
```

SYNOPSIS

```
delete_file_at_startup { filename }
```

FUNCTION

```
This procedure will delete every file added to the file
$CHECK_TESTSUITE_ROOT/.testsuite_delete on the startup of a testrun
```

INPUTS

```
filename - full path file name of file to add to
           $CHECK_TESTSOUTE_ROOT/.testsuite_delete file
```

RESULT

```
no results
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 3.5 [file_procedures delete_directory], page 42.

3.8 get_binary_path

NAME

```
get_binary_path -- get host specific binary path
```

SYNOPSIS

```
get_binary_path { hostname binary }
```

FUNCTION

```
This procedure will parse the binary-path.conf configuration file of the
testsuite. In this file the user can configure his host specific binary
path names.
```

INPUTS

```
hostname - hostname where a binary should be found
binary   - binary name (e.g. expect)
```

RESULT

```
The full path name of the binary on the given host. The return value
depends on the entries in the binary-path.conf file.
```

EXAMPLE

```
???
```

NOTES

```
The binary-path.conf file has following syntax:
Each line has 3  entries:
hostname binary path. The $ARCH variable is resolved.
```

BUGS

```
???
```

SEE ALSO

See Section 3.9 [file_procedures get_dir_names], page 44.

3.9 get_dir_names

NAME

```
get_dir_names -- return all subdirectory names
```

SYNOPSIS

```
get_dir_names { path }
```

FUNCTION

```
read in directory and return a list of subdirectory names
```

INPUTS

```
path - path to read in
```

RESULT

```
list of subdirectory names
```

EXAMPLE

```
set dirs [ get_dir_names /tmp ]
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 3.10 [file_procedures get_file_names], page 45.

3.10 get_file_names

NAME

```
get_file_names -- return all file names of directory
```

SYNOPSIS

```
get_file_names { path {ext "*"} }
```

FUNCTION

```
read in directory and return a list of file names in this directory
```

INPUTS

```
path - path to read in (directory)
ext  - file extension (default "*")
```

RESULT

```
list of file names
```

EXAMPLE

```
set files [ get_file_names /tmp ]
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 3.9 [file_procedures get_dir_names], page 44.

3.11 test_file

NAME

```
test_file -- test procedure
```

SYNOPSIS

```
test_file { me two }
```

FUNCTION

```
this function is just for test the correct function call
```

INPUTS

```
me  - first output parameter
two - second output parameter
```

RESULT

```
output to stdout:
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

3.12 wait_for_file

NAME

```
wait_for_file -- wait for file to appear/dissappear/...
```

SYNOPSIS

```
wait_for_file { path_to_file seconds { to_go_away 0 }
{ do_error_check 1 } }
```

FUNCTION

```
Wait a given number of seconds fot the creation or deletion of a file.
```

INPUTS

```
path_to_file         - full path file name of file
seconds              - timeout in seconds
{ to_go_away 0 }     - flag, (0=wait for creation, 1 wait for deletion)
{ do_error_check 1 } - flag, (0=do not report errors, 1 report errors)
```

RESULT

```
-1 for an unsuccessful waiting, 0 no errors
```

SEE ALSO

See Section 3.5 [file_procedures delete_directory], page 42.
See Section 18.68 [sge_procedures wait_for_load_from_all_queues], page 215.
See Section 3.12 [file_procedures wait_for_file], page 46.
See Section 18.67 [sge_procedures wait_for_jobstart], page 214.
See Section 18.64 [sge_procedures wait_for_end_of_transfer], page 212.
See Section 18.66 [sge_procedures wait_for_jobpending], page 213.
See Section 18.65 [sge_procedures wait_for_jobend], page 213.

4 install_core_system

4.1 get_spool_dir

NAME

```
get_spool_dir -- ???
```

SYNOPSIS

```
get_spool_dir { host subdir }
```

FUNCTION

```
???
```

INPUTS

```
host   - ???
subdir - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

4.2 install_execd

NAME

```
install_execd -- ???
```

SYNOPSIS

```
install_execd { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.3 install_qmaster

NAME

```
install_qmaster -- ???
```

SYNOPSIS

```
install_qmaster { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.4 kill_running_system

NAME

```
kill_running_system -- ???
```

SYNOPSIS

```
kill_running_system { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.5 read_install_list

NAME

```
read_install_list -- ???
```

SYNOPSIS

```
read_install_list { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.6 setup_check_user_permissions

NAME

```
setup_check_user_permissions -- ???
```

SYNOPSIS

```
setup_check_user_permissions { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.7 setup_conf

NAME

```
setup_conf -- ???
```

SYNOPSIS

```
setup_conf { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.8 setup_deadlineuser

NAME

```
setup_deadlineuser -- ???
```

SYNOPSIS

```
setup_deadlineuser { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.9 setup_default_calendars

NAME

```
setup_default_calendars -- ???
```

SYNOPSIS

```
setup_default_calendars { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.10 setup_inhouse_cluster

NAME

```
setup_inhouse_cluster -- ???
```

SYNOPSIS

```
setup_inhouse_cluster { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.11 setup_mytestpe

NAME

```
setup_mytestpe -- ???
```

SYNOPSIS

```
setup_mytestpe { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.12 setup_mytestproject

NAME

```
setup_mytestproject -- ???
```

SYNOPSIS

```
setup_mytestproject { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.13 setup_queues

NAME

```
setup_queues -- ???
```

SYNOPSIS

```
setup_queues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.14 setup_schedconf

NAME

```
setup_schedconf -- ???
```

SYNOPSIS

```
setup_schedconf { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.15 setup_testcheckpointobject

NAME

```
setup_testcheckpointobject -- ???
```

SYNOPSIS

```
setup_testcheckpointobject { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

4.16 write_install_list

NAME

```
write_install_list -- ???
```

SYNOPSIS

```
write_install_list { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

5 loadcheck

5.1 check_numb_proc

NAME

```
check_numb_proc -- ???
```

SYNOPSIS

```
check_numb_proc { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

5.2 get_numb_proc

NAME

```
get_numb_proc -- ???
```

SYNOPSIS

```
get_numb_proc { hostname }
```

FUNCTION

```
???
```

INPUTS

```
hostname - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

6 migrate

6.1 init_level

NAME

```
init_level -- ???
```

SYNOPSIS

```
init_level { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

6.2 shadowd_cleanup

NAME

```
shadowd_cleanup -- ???
```

SYNOPSIS

```
shadowd_cleanup { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

6.3 shadowd_kill_all_shadowd

NAME

```
shadowd_kill_all_shadowd -- ???
```

SYNOPSIS

```
shadowd_kill_all_shadowd { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

6.4 shadowd_kill_master_and_sheduler

NAME

```
shadowd_kill_master_and_sheduler -- ???
```

SYNOPSIS

```
shadowd_kill_master_and_sheduler { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

6.5 shadowd_kill_shadowd_master_and_shadowd_sheduler

NAME

`shadowd_kill_shadowd_master_and_shadowd_sheduler -- ???`

SYNOPSIS

`shadowd_kill_shadowd_master_and_shadowd_sheduler { }`

FUNCTION

???

RESULT

???

EXAMPLE

???

NOTES

???

BUGS

???

SEE ALSO

See '/'

6.6 shadowd_setup

NAME

`shadowd_setup -- ???`

SYNOPSIS

`shadowd_setup { }`

FUNCTION

???

RESULT

???

EXAMPLE

???

NOTES

???

BUGS

???

SEE ALSO

See '/'

6.7 shadowd_startup

NAME

```
shadowd_startup -- ???
```

SYNOPSIS

```
shadowd_startup { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

6.8 shadowd_wait_for_startup

NAME

```
shadowd_wait_for_startup -- ???
```

SYNOPSIS

```
shadowd_wait_for_startup { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

7 migration

7.1 calendarclear_queue

NAME

```
calendarclear_queue -- ???
```

SYNOPSIS

```
calendarclear_queue { queue_list }
```

FUNCTION

```
???
```

INPUTS

```
queue_list - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.2 calendardisable_queue

NAME

```
calendardisable_queue -- ???
```

SYNOPSIS

```
calendardisable_queue { queue_list }
```

FUNCTION

```
???
```

INPUTS

```
queue_list - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.3 calendarsuspend_queue

NAME

```
calendarsuspend_queue -- ???
```

SYNOPSIS

```
calendarsuspend_queue { queue_list }
```

FUNCTION

```
???
```

INPUTS

```
queue_list - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.4 check_calendardisable_migration_on_slavequeue_suspend

NAME

```
check_calendardisable_migration_on_slavequeue_suspend -- ???
```

SYNOPSIS

```
check_calendardisable_migration_on_slavequeue_suspend { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.5 check_calendardisable_migration_on_slavequeue_threshold_suspend

NAME

```
check_calendardisable_migration_on_slavequeue_threshold_suspend -- ???
```

SYNOPSIS

```
check_calendardisable_migration_on_slavequeue_threshold_suspend { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.6 check_calendarsuspend_master_migration

NAME

```
check_calendarsuspend_master_migration -- ???
```

SYNOPSIS

```
check_calendarsuspend_master_migration { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.7 check_calendarsuspend_slave_migration

NAME

`check_calendarsuspend_slave_migration -- ???`

SYNOPSIS

`check_calendarsuspend_slave_migration { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See '/'

7.8 check_master_migration

NAME

`check_master_migration -- ???`

SYNOPSIS

`check_master_migration { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See '/'

7.9 check_slave_migration

NAME

```
check_slave_migration -- ???
```

SYNOPSIS

```
check_slave_migration { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

7.10 clean_up_checkpoint_job

NAME

```
clean_up_checkpoint_job -- ???
```

SYNOPSIS

```
clean_up_checkpoint_job { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

7.11 clean_up_checkpointing

NAME

```
clean_up_checkpointing -- ???
```

SYNOPSIS

```
clean_up_checkpointing { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.12 clean_up_pe

NAME

```
clean_up_pe -- ???
```

SYNOPSIS

```
clean_up_pe { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.13 clean_up_queues

NAME

```
clean_up_queues -- ???
```

SYNOPSIS

```
clean_up_queues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.14 setup_checkpointing

NAME

```
setup_checkpointing -- ???
```

SYNOPSIS

```
setup_checkpointing { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.15 setup_pe

NAME

```
setup_pe -- ???
```

SYNOPSIS

```
setup_pe { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.16 setup_queues

NAME

```
setup_queues -- ???
```

SYNOPSIS

```
setup_queues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

7.17 start_checkpoint_job

NAME

```
start_checkpoint_job -- ???
```

SYNOPSIS

```
start_checkpoint_job { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

7.18 threshold_suspend_queue

NAME

```
threshold_suspend_queue -- ???
```

SYNOPSIS

```
threshold_suspend_queue { queue_list }
```

FUNCTION

```
???
```

INPUTS

```
queue_list - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

7.19 threshold_suspend_queue_clear

NAME

```
threshold_suspend_queue_clear -- ???
```

SYNOPSIS

```
threshold_suspend_queue_clear { queue_list }
```

FUNCTION

```
???
```

INPUTS

```
queue_list - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

8 parser

8.1 output_array

NAME

```
output_array -- ???
```

SYNOPSIS

```
output_array { input }
```

FUNCTION

```
???
```

INPUTS

```
input - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

8.2 overview

NAME

```
Parsing Functions -- parsing and processing of different input formats
```

SYNOPSIS

```
source parser.tcl
# call parsing functions
```

FUNCTION

```
The tcl library file parser.tcl provides a set of functions for
parsing and processing of input data coming for example from the
execution of programs like ps, qstat, qacct etc.

The parsing functions take the input, apply certain filtering and
processing steps, and provide as output a uniform representation
of the data in a TCL array.

The following filtering/processing steps can be done:
   - Replacements:
     By this mechanism certain defined field contents can be replaced
     by other values. This may be needed for later processing steps.
     Example: Output of qstat -ext contains "NA" in the columns cpu,
              mem and io when online accounting information is not yet
              available. To be able to do computations on such a column,
              the value "NA" can be automatically replaced by the value
              "0" during the parsing step.

   - Transformations:
     Transformations can be performed on the data of certain defined
     columns to change the data representation of the values.
     Example: The output of qstat -ext contains the values for cpu
              usage in the format "days:hours:minutes:seconds". To be
              able to do computations on cpu values, it is necessary
              to transform the given representation to a numerical
              value in seconds.

              Date and Time is often given in a textual representation.
              To do computations on date/time values, e.g. compute
              the time period between a start and an end timestamp, it
              is usefull to transform the date/time data to a UNIX-
              timestamp.

   - Rules to handle multiple records for one output unit:
     Often one record in the output array is built out of different
     records in the input data. In this case, data values have to be
     combined following a certain rule.
     Example: The information given by qacct for a parallel job shall
              be output in one record. The resource values (cpu, mem and i
              shall be summed up, the involved queues shall be returned
              as a list, ...
```

EXAMPLES

```
Examples are given in the documentation of the different parsing
functions.
Also the functions parse_qstat and parse_qacct are a good example
for the usage of the parsing functions.
```

SEE ALSO

See Section 8.6 [parser parse_fixed_column_lines], page 74.
See Section 8.9 [parser process_named_record], page 77.
See Section 8.10 [parser process_output_array], page 80.
See Section 8.3 [parser overview_parsing_replacements], page 71.
See Section 8.5 [parser overview_parsing_transformations], page 73.
See Section 8.4 [parser overview_parsing_rules], page 72.

8.3 overview_parsing_replacements

NAME

```
Parsing Replacements -- automatic replacement of certain cell contents
```

SYNOPSIS

```
set replace(<column/field>,<contents>) value
```

FUNCTION

```
For processing of data tables or records, it is sometimes necessary
to replace certain contents or to add missing contents.

Parsing Functions of this module allow the specification of a TCL array
describing replacement rules that will be automatically evaluated
during the parsing of input data.

Example:
   If a numerical value is not yet known, its value is reported as "NA".
   The occurence of "NA" in a table cell prohibits doing calculations
   including this cell.
   Therefor it shall be replaced by "0".
```

EXAMPLE

```
# Value NA in cells of column 1 shall be replaced by 0
set replace(1,NA) 0

# Missing values for record field "location" shall be replaced by "unknown
set replace(location,) unknown
```

SEE ALSO

See Section 8.6 [parser parse_fixed_column_lines], page 74.
See Section 8.9 [parser process_named_record], page 77.

8.4 overview_parsing_rules

NAME

```
Parsing Rules -- Rules to combine multiple values
```

SYNOPSIS

```
set rules(field/column) functionname
```

FUNCTION

```
If an input table contains multiple rows that shall be combined into
one row in the output table, the data must be combined following certain
rules.
Therefor the processing functions in this module allow the specification
of rules that are applied to cells of certain table columns or record
fields.

The processing functions evaluate the following TCL expression:
eval $rules(field/column) present_output_value new_output_value

The functions representing a rule must be prepared to accept
two input values and return one combined output value.

The following rules are contained in this module:
   rule_list:
      Return a list containing the elements of both input values.
   rule_sum:
      Calculate the sum of the two input values.

   rule_min:
      Return the smaller of the two input values.

   rule_max:
      Return the greater of the two input values.
```

EXAMPLE

```
set rules(5) rule_sum
set rules(start_time) rule_min
set rules(taskid) rule_list
```

SEE ALSO

See Section 8.10 [parser process_output_array], page 80.
See Section 8.9 [parser process_named_record], page 77.

8.5 overview_parsing_transformations

NAME

```
Parsing Transformations -- tranformation of contents to other format
```

SYNOPSIS

```
set transform(column/field) expression
```

FUNCTION

```
To be able to process field or table cell contents it is often necessary
to change the data representation of the contents.

Parsing Functions of this module allow the specification of a TCL array
describing transformation rules that will be automatically evaluated
during the parsing of input data.

The parsing functions process the following TCL expression:
eval $transform(column/field) value

The specified transformation expression must be prepared to accept
exactly one parameter and return the transformed value.

Example:
   To do calculations on date/time values, it is usefull to transform
   their data representation from text format to UNIX-Timestamp.

The following transformation functions are provided in this module:
   transform_duration:
      Transform a duration given as days:hours:minutes:seconds
      where hour, minutes, seconds are written with leading 0 where
      necessary to an integer representing the duration in seconds.
   transform_date_time:
      Transform a textual representation of date/time to a
      UNIX timestamp (seconds since 01/01/1970).
      The textual representation must follow the rules defined in the
      manual pages for the TCL command "clock scan".
```

EXAMPLE

```
set transform(start_time) transform_date_time
```

SEE ALSO

See Section 8.6 [parser parse_fixed_column_lines], page 74.

See Section 8.9 [parser process_named_record], page 77.

8.6 parse_fixed_column_lines

NAME

```
parse_fixed_column_lines -- parse fixed size input table
```

SYNOPSIS

```
parse_fixed_column_lines input output position
                         [start_line] [replace] [transform]
```

FUNCTION

```
Parses an input table given as string in variable input with the following
format:
  - table rows are separated by newline (\n)
  - table columns have fixed width
The result is stored in a TCL array, the indices have the form
<row>,<column>, e.g. "0,4"; the first row or column has number 0, so
table indicese range from "0,0" to "n,m".
Header lines may be stripped by specifying a start_line > 0.
Certain contents of cells can be replaced, e.g. if a numerical cell
is empty (string ""), it could be set to 0.
A transformation can be performed while parsing the input, e.g. formatted
date/time can be transformed to UNIX timestamp.
Rules for replacement and transformation can be set per column.
In addition to the table cells, two entries are set in the output array
describing the tables dimensions: output(rows) and output(cols).
```

INPUTS

```
The parameters input, output, position, replace and transform are
passed by reference.

input        - name of the string variable containing the input table
output       - name of the output variable in which to place the resulting
               TCL array
position     - name of the TCL array containing the positioning informatio
               Contains one entry per column of the input table in the form
               "<start_position> <end_position>" where start_position and
               end position are valid index parameters to the TCL function
               "string range". Example: "0 5" or "70 end".
               The array is indexed by the column number starting at 0 for
```

```
                 first column, e.g. set position(0) "0 5".
[start_line] - line from which to start reading the table (default 0 = firs
[replace]    - name of the TCL array containing rules to replace certain
                 cell contents - if parameter is not passed to function, no r
                 will be made.
                 The index of the array is build as <column_number>,<string_t
                 the arrays values are the strings that replace any occurence
                 string_to_replace in column column_number.
                 Example: set replace(0,) -1 sets each empty cell in row 0 to
                          set replace(0,NA) -1 sets each cell containing NA i
[transform]  - name of the TCL array containing rules to transform the cont
                 certain cells - if parameter is not passed to function, no t
                 will be made.
                 The array is indexed by the column number starting at 0 for
                 first column, e.g. set transform(2) transform_date_time.
                 The value of an array entry is a tcl command that is called
                 a cells value as parameter and returns the new value.
```

RESULT

```
output - The resulting TCL array is placed in the variable that is referenc
         the parameter output in the callers namespace.
```

EXAMPLE

```
source parser.tcl

set input "id num date
a 1 10/30/2000
a 2 10/31/2000
b 5 11/17/2000
- 8 01/05/2000"

set position(0) "0 0"
set position(1) "2 2"
set position(2) "4 13"

set replace(0,-) ?

set transform(2) transform_date_time

parse_fixed_column_lines input output position 1 replace transform

output_array output

Result:
a       1       972860400
a       2       972946800
b       5       974415600
?       8       947026800
```

NOTES

```
The output of parse_fixed_column_lines will usually be postprocessed
by the function process_output_array.
The function repeat_columns can be used to fill in missing information
```

```
into the output table of parse_fixed_column_lines.
```

SEE ALSO

See ‘parser/repeat_columns’
See Section 8.10 [parser process_output_array], page 80.
See Section 8.3 [parser overview_parsing_replacements], page 71.
See Section 8.5 [parser overview_parsing_transformations], page 73.

8.7 parse_qacct

NAME

```
parse_qacct -- parse information from qacct command
```

SYNOPSIS

```
parse_qacct input output [jobid]
```

FUNCTION

```
The function parses the output given from a qacct -j <jobid> command
and returns the information in a TCL array indexed by the fieldnames.
The following processing is applied to the data:
   - taskids "unknown" are replaced by "1"
   - Date/Time is transformed to UNIX timestamp
If multiple records are combined into one output record
   - queuenames, hostnames, stati and taskid's are appended as lists
   - resource values are summed up
   - submit and starttime are the minimum of all values
   - end time is the maximum of all values
```

INPUTS

```
input   - name of a string variable containing the output of qacct
output  - TCL array in which to store the results
[jobid] - jobid that was used for qacct command
```

RESULT

```
The output array is filled with the processed data.
If a jobid was specified, the array is indexed by the fieldnames,
if not, the index is built as "jobid,fieldname".
```

8.8 parse_qstat

NAME

```
parse_qstat -- parse output of a qstat [-ext] command
```

SYNOPSIS

```
parse_qstat input output [jobid] [ext]
```

FUNCTION

```
Parses the output of a qstat or (in SGEEE) qstat -ext command.
If a certain jobid is specified, only the information for
this job is returned, otherwise information for all jobs.
```

```
The following processing is applied to data:
   - numerical information containing empty strings or NA
     is set to 0
   - durations and data/time strings are transformed to
     UNIX timestamp

The following rules are applied to the data, if multiple values
have to be combined into one:
   - take the minimum of submit/start times
   - sum up all sort of resource values, tickets etc.
   - build lists from qnames, task category (MASTER/SLAVE)
     and taskid's
```

INPUTS

```
input   - name of the input string with data from qstat command
output  - name of the array in which to return results
[jobid] - jobid for filtering a certain job
[ext]   - 0: qstat command, 1: qstat -ext command
```

RESULT

```
The TCL array output is filled with the processed data.
If a certain jobid is specified, the arrays index consists of
the columnnames (e.g. id, prior), if no jobid is specified,
the index has the form "jobid,columnname" (e.g. 182,id).
```

8.9 process_named_record

NAME

```
process_named_record -- parse records with named elements
```

SYNOPSIS

```
process_named_record input output delimiter index \
                     [id] [head_line] [tail_line] \
                     [replace] [transform] [rules]
```

FUNCTION

```
Parses input data in the form of records that
  - contains a tuple <field_name><whitespace><field_value> in each line
  - records are separated by a fixed record delimiter

The records are stored in an TCL associative array, from which record fiel
the index is created can be specified in a parameter.

Records can be filtered by the contents of any fields contained in the ind
field list.

Heading or trailing lines can be excluded from parsing.

Certain input field values can be replaced by specifying a replace rule
per field name.
```

```
Input field values can be transformed by specifying a transformation rule
per field name, it is for example possible to convert formatted date/time
to UNIX timestamp during the parsing of the input.

If multiple records exist for one index value, a rule can be specified how
merge the values, e.g. sum, average, build a list etc.
```

INPUTS

```
The parameters input, output, replace, transform and rules are
passed by reference.

input       - name of a string variable containing the input
output      - name of a TCL array into which the output is written
delimiter   - record delimiter (one line)
index       - list of fieldnames building the index
[id]        - list of fieldvalues refering to the index. Only records
              containing these field values will be processed.
[head_line] - number of lines to skip at the beginning of input
[tail_line] - number of lines to skip at the end of input
[replace]   - name of the TCL array containing rules to replace certain
              field contents - if parameter is not passed to function, no r
              will be made.
              The index of the array is build as <field_name>,<string_to_re
              the arrays values are the strings that replace any occurence
              string_to_replace in column column_number.
              Example: set replace(jobname,) noname sets each empty field w
                       set replace(cpu,NA) 0 sets each field with name cpu
[transform] - name of the TCL array containing rules to transform the conte
              certain cells - if parameter is not passed to function, no tr
              will be made.
              The array is indexed by the field name.
              The value of an array entry is a tcl command that is called w
              a cells value as parameter and returns the new value.
[rules]     - name of a TCL array containing rules to apply to field values
              if multiple records have the same index.
              The value of an array entry is the name of a TCL function tha
              is called and is passed as parameters the value of the corres
              entry in the output array and the new value in the actual rec
              If no rule is set for a field, a new value replaces the old o
```

RESULT

```
output - Name of a TCL array in which to place the resulting records.
```

EXAMPLE

```
source parser.tcl

proc output_result {output} {
   upvar $output out

   puts [format "%8s %-12s %-12s %-25s %8s" jobid task(s) jobname queue(s)
   if { $out(index) == "" } {
      puts [format "%8d %-12s %-12s %-25s %8d" $out(jobid) $out(taskid) $ou
   } else {
      foreach i $out(index) {

         puts [format "%8d %-12s %-12s %-25s %8d" $out(${i}jobid) $out(${i
      }
   }
}

set input "some header line
jobid    123
taskid   1
jobname  sleeper.sh
queue    balrog.q
cpu      0:00:00:02
-------
jobid    124
taskid   1
jobname  worker.sh
queue    sowa.q
cpu      0:00:01:00
-------
jobid    124
taskid   2
jobname  worker.sh
queue    elendil.q
cpu      0:00:00:55
-------
jobid    124
taskid   3
jobname  worker.sh
queue    balrog.q
cpu      NA
==========================
some trailing garbage ...
in multiple lines
"

set replace(cpu,NA) "0:00:00:00"
set transform(cpu)  transform_cpu
set rules(taskid)    rule_list
set rules(queue)     rule_list
set rules(cpu)       rule_sum

# show all jobs, one record per jobid (means: join taskid's)
unset output
process_named_record input output "-------" "jobid" "" 1 3 replace transfo
output_result output

Result:
   jobid task(s)      jobname      queue(s)                       cpu
     123 1            sleeper.sh   balrog.q                         2
     124 1 2 3        worker.sh    sowa.q elendil.q balrog.q      115
```

```
# show all jobs, one record for each taskid
unset output
process_named_record input output "-------" "jobid taskid" "" 1 3 replace t
output_result output

Result:
   jobid task(s)      jobname     queue(s)                      cpu
     123 1            sleeper.sh  balrog.q                        2
     124 1            worker.sh   sowa.q                         60
     124 2            worker.sh   elendil.q                      55
     124 3            worker.sh   balrog.q                        0

# show job 123
unset output
process_named_record input output "-------" "jobid" "123" 1 3 replace trans
output_result output

Result:
   jobid task(s)      jobname     queue(s)                      cpu
     123 1            sleeper.sh  balrog.q                        2

# show job 124, task 2
unset output
process_named_record input output "-------" "jobid taskid" "124 2" 1 3 repl
output_result output

Result:
   jobid task(s)      jobname     queue(s)                      cpu
     124 2            worker.sh   elendil.q                      55

# show all jobs that ran in queue balrog.q, one record per jobid
unset output
process_named_record input output "-------" "queue jobid" "balrog.q" 1 3 re
output_result output

Result:
   jobid task(s)      jobname     queue(s)                      cpu
     123 1            sleeper.sh  balrog.q                        2
     124 3            worker.sh   balrog.q                        0
```

SEE ALSO

See Section 8.3 [parser overview_parsing_replacements], page 71.
See Section 8.5 [parser overview_parsing_transformations], page 73.
See Section 8.4 [parser overview_parsing_rules], page 72.

8.10 process_output_array

NAME

```
process_output_array -- postprocessing of tables
```

SYNOPSIS

```
process_output_array input output names [id] [rules]
```

FUNCTION

```
The function takes a input a TCL array containing a
data table indexed by "row,column".
It applies filtering and rules for the combination of
multiple rows and outputs a TCL array indexed by the
first column of the input table (optionally) and the column names
given in the parameter "names".
```

INPUTS

```
The parameters input, output, names and rules are
passed by reference.

input   - name of a TCL array containing the input
output  - name of a TCL array for the output
names   - name of a TCL array containing the column names; it is indexed
          by the column number starting with 0
[id]    - optional value of cells in column 0 by which filtering is done.
          If it's value is != "", only rows that have the value $id in
          the first column are processed.
          If id is not passed or its value is a string of length 0, all
          rows from the input array are processed, the indexes in the
          output array are prefixed by the contents of column 0 from the
          input array.
[rules] - Rules to apply on values of cells, if multiple rows exist
          with the same value in the index column 0.
          A rule is a TCL expression that gets two parameters: the present
          value of the output array for the specific index and the new
          value of the actually parsed row.
          For each column of the input table a rule can be defined, identi
          by the column number as index of the array rules.
          If no rule is specified for a column, new values will replace th
          present values.
```

RESULT

```
output - The resulting TCL array is placed in the variable that is referen
         the parameter output in the callers namespace.
```

EXAMPLE

```
# Take the result of example for function parse_fixed_column_lines
a       1       972860400
a       2       972946800
b       5       974415600
?       8       947026800

proc output_result {output} {
   upvar $output out

   puts [format "%-5s %-10s %s" "id" "task(s)" "date"]
   foreach i $out(index) {
      puts [format "%-5s %-10s %s" $out(${i}id) $out(${i}task) [clock form
   }
}

set names(0) id
set names(1) task          ; set rules(1) rule_list
set names(2) start_date    ; set rules(2) rule_min

process_output_array output newoutput names "" rules
puts [array names newoutput] ; output_result newoutput
Result:
index a,task a,start_date b,id id ?,id b,task b,start_date a,id task start_
id    task(s)    date
a     1 2        Mon Oct 30 00:00:00 MET 2000
b     5          Fri Nov 17 00:00:00 MET 2000
?     8          Wed Jan 05 00:00:00 MET 2000

process_output_array output newoutput names a rules
puts [array names newoutput] ; output_result newoutput
Result:
index id start_date task
id    task(s)    date
a     1 2        Mon Oct 30 00:00:00 MET 2000
```

SEE ALSO

See Section 8.6 [parser parse_fixed_column_lines], page 74.
See Section 8.4 [parser overview_parsing_rules], page 72.

8.11 repeat_column

NAME

```
repeat_column -- repeat column contents where missing
```

SYNOPSIS

```
repeat_column input [column]
```

FUNCTION

```
Processes a table stored in a TCL array (e.g. output from
parse_fixed_column_lines) and repeats values of cells where
they are missing in the following rows.
Example: Qstat output for parallel jobs outputs the jobid
only for the first task of the job in a certain queue, the
following tasks of this job in the same queue are listed
without jobid. For easier processing of the job table,
it is necessary to fill in the missing jobid's.
```

INPUTS

```
input    - TCL array containing a table, array indexes have the
           form "row,column", e.g. "10,5"
[column] - column number in which to repeat missing values,
           default is column 0
```

RESULT

```
Table in TCL array input is changed
```

SEE ALSO

See Section 8.6 [parser parse_fixed_column_lines], page 74.

8.12 rule_list

NAME

```
rule_list -- ???
```

SYNOPSIS

```
rule_list { a b }
```

FUNCTION

```
???
```

INPUTS

```
a - ???
b - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

8.13 rule_max

NAME

```
rule_max -- ???
```

SYNOPSIS

```
rule_max { a b }
```

FUNCTION

```
???
```

INPUTS

```
a - ???
b - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

8.14 rule_min

NAME

```
rule_min -- ???
```

SYNOPSIS

```
rule_min { a b }
```

FUNCTION

```
???
```

INPUTS

```
a - ???
b - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

8.15 rule_sum

NAME

```
rule_sum -- ???
```

SYNOPSIS

```
rule_sum { a b }
```

FUNCTION

```
???
```

INPUTS

```
a - ???
b - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

8.16 transform_cpu

NAME

```
transform_cpu -- ???
```

SYNOPSIS

```
transform_cpu { s_cpu }
```

FUNCTION

```
???
```

INPUTS

```
s_cpu - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

8.17 transform_date_time

NAME

```
transform_date_time -- ???
```

SYNOPSIS

```
transform_date_time { value }
```

FUNCTION

```
???
```

INPUTS

```
value - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

9 performance

9.1 cleanup_queues

NAME

```
cleanup_queues -- ???
```

SYNOPSIS

```
cleanup_queues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

NAME

```
cleanup_queues -- ???
```

SYNOPSIS

```
cleanup_queues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

NAME

```
cleanup_queues -- ???
```

SYNOPSIS

```
cleanup_queues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

9.2 do_perform_test

NAME

```
do_perform_test -- ???
```

SYNOPSIS

```
do_perform_test { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

9.3 init_level

NAME

```
init_level -- ???
```

SYNOPSIS

```
init_level { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

NAME

```
init_level -- ???
```

SYNOPSIS

```
init_level { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

NAME

```
init_level -- ???
```

SYNOPSIS

```
init_level { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

9.4 performance_test

NAME

```
performance_test -- ???
```

SYNOPSIS

```
performance_test { job_count_loops job_run_loops }
```

FUNCTION

```
???
```

INPUTS

```
job_count_loops - ???
job_run_loops   - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

9.5 setup_queues

NAME

```
setup_queues -- ???
```

SYNOPSIS

```
setup_queues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

NAME

```
setup_queues -- ???
```

SYNOPSIS

```
setup_queues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

NAME

```
setup_queues -- ???
```

SYNOPSIS

```
setup_queues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

9.6 submit_jobs

NAME

```
submit_jobs -- ???
```

SYNOPSIS

```
submit_jobs { job_count job_time }
```

FUNCTION

```
???
```

INPUTS

```
job_count - ???
job_time  - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10 qalter

10.1 qalter_A

NAME

```
qalter_A -- ???
```

SYNOPSIS

```
qalter_A { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.2 qalter_M

NAME

```
qalter_M -- ???
```

SYNOPSIS

```
qalter_M { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.3 qalter_N

NAME

```
qalter_N -- ???
```

SYNOPSIS

```
qalter_N { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.4 qalter_P

NAME

```
qalter_P -- ???
```

SYNOPSIS

```
qalter_P { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.5 qalter_S

NAME

```
qalter_S -- ???
```

SYNOPSIS

```
qalter_S { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.6 qalter_V

NAME

```
qalter_V -- ???
```

SYNOPSIS

```
qalter_V { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.7 qalter_a

NAME

```
qalter_a -- ???
```

SYNOPSIS

```
qalter_a { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.8 qalter_ac

NAME

```
qalter_ac -- ???
```

SYNOPSIS

```
qalter_ac { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.9 qalter_c

NAME

```
qalter_c -- ???
```

SYNOPSIS

```
qalter_c { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.10 qalter_ckpt

NAME

```
qalter_ckpt -- ???
```

SYNOPSIS

```
qalter_ckpt { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.11 qalter_clear

NAME

```
qalter_clear -- ???
```

SYNOPSIS

```
qalter_clear { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.12 qalter_cwd

NAME

```
qalter_cwd -- ???
```

SYNOPSIS

```
qalter_cwd { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.13 qalter_dc

NAME

```
qalter_dc -- ???
```

SYNOPSIS

```
qalter_dc { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.14 qalter_e

NAME

```
qalter_e -- ???
```

SYNOPSIS

```
qalter_e { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.15 qalter_hard

NAME

```
qalter_hard -- ???
```

SYNOPSIS

```
qalter_hard { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.16 qalter_hold

NAME

```
qalter_hold -- ???
```

SYNOPSIS

```
qalter_hold { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.17 qalter_j

NAME

```
qalter_j -- ???
```

SYNOPSIS

```
qalter_j { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.18 qalter_l

NAME

```
qalter_l -- ???
```

SYNOPSIS

```
qalter_l { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.19 qalter_m

NAME

```
qalter_m -- ???
```

SYNOPSIS

```
qalter_m { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.20 qalter_notify

NAME

```
qalter_notify -- ???
```

SYNOPSIS

```
qalter_notify { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.21 qalter_o

NAME

```
qalter_o -- ???
```

SYNOPSIS

```
qalter_o { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.22 qalter_p

NAME

```
qalter_p -- ???
```

SYNOPSIS

```
qalter_p { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.23 qalter_pe

NAME

`qalter_pe -- ???`

SYNOPSIS

`qalter_pe { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See ‘/’

10.24 qalter_q

NAME

`qalter_q -- ???`

SYNOPSIS

`qalter_q { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See ‘/’

10.25 qalter_qs_args

NAME

```
qalter_qs_args -- ???
```

SYNOPSIS

```
qalter_qs_args { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.26 qalter_rn

NAME

```
qalter_rn -- ???
```

SYNOPSIS

```
qalter_rn { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.27 qalter_ry

NAME

```
qalter_ry -- ???
```

SYNOPSIS

```
qalter_ry { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.28 qalter_sc

NAME

```
qalter_sc -- ???
```

SYNOPSIS

```
qalter_sc { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.29 qalter_soft

NAME

```
qalter_soft -- ???
```

SYNOPSIS

```
qalter_soft { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.30 qalter_v

NAME

```
qalter_v -- ???
```

SYNOPSIS

```
qalter_v { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.31 qalter_verify

NAME

```
qalter_verify -- ???
```

SYNOPSIS

```
qalter_verify { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.32 qalter_w

NAME

```
qalter_w -- ???
```

SYNOPSIS

```
qalter_w { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

10.33 run_dummy_jobs

NAME

```
run_dummy_jobs -- ???
```

SYNOPSIS

```
run_dummy_jobs { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

10.34 start_testjob

NAME

```
start_testjob -- ???
```

SYNOPSIS

```
start_testjob { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11 qconf

11.1 check_exec_conf

NAME

```
check_exec_conf -- ???
```

SYNOPSIS

```
check_exec_conf { host_list attr_name check_value }
```

FUNCTION

```
???
```

INPUTS

```
host_list   - ???
attr_name   - ???
check_value - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11.2 check_queue_conf

NAME

```
check_queue_conf -- ???
```

SYNOPSIS

```
check_queue_conf { queue_list attr_name check_value }
```

FUNCTION

```
???
```

INPUTS

```
queue_list  - ???
attr_name   - ???
check_value - ???
```

RESULT

```
???
```

EXAMPLE

    ???

NOTES

    ???

BUGS

    ???

SEE ALSO

See '/'

11.3 init_level

NAME

    init_level -- ???

SYNOPSIS

    init_level { }

FUNCTION

    ???

RESULT

    ???

EXAMPLE

    ???

NOTES

    ???

BUGS

    ???

SEE ALSO

See '/'

11.4 qconf_Aattr_check

NAME

    qconf_Aattr_check -- ???

SYNOPSIS

    qconf_Aattr_check { }

FUNCTION

    ???

RESULT

    ???

EXAMPLE

    ???

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11.5 qconf_Dattr_check

NAME

```
qconf_Dattr_check -- ???
```

SYNOPSIS

```
qconf_Dattr_check { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11.6 qconf_Mattr_check

NAME

```
qconf_Mattr_check -- ???
```

SYNOPSIS

```
qconf_Mattr_check { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11.7 qconf_Rattr_check

NAME

```
qconf_Rattr_check -- ???
```

SYNOPSIS

```
qconf_Rattr_check { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11.8 qconf_aattr_check

NAME

```
qconf_aattr_check -- ???
```

SYNOPSIS

```
qconf_aattr_check { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11.9 qconf_addqueues

NAME

```
qconf_addqueues -- ???
```

SYNOPSIS

```
qconf_addqueues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11.10 qconf_dattr_check

NAME

```
qconf_dattr_check -- ???
```

SYNOPSIS

```
qconf_dattr_check { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11.11 qconf_mattr_check

NAME

```
qconf_mattr_check -- ???
```

SYNOPSIS

```
qconf_mattr_check { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11.12 qconf_rattr_check

NAME

```
qconf_rattr_check -- ???
```

SYNOPSIS

```
qconf_rattr_check { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

11.13 qconf_removequeues

NAME

```
qconf_removequeues -- ???
```

SYNOPSIS

```
qconf_removequeues { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12 qdel

12.1 are_jobs_deleted

NAME

```
are_jobs_deleted -- ???
```

SYNOPSIS

```
are_jobs_deleted { job_list }
```

FUNCTION

```
???
```

INPUTS

```
job_list - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12.2 init_level

NAME

```
init_level -- ???
```

SYNOPSIS

```
init_level { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12.3 qdel_all

NAME

```
qdel_all -- ???
```

SYNOPSIS

```
qdel_all { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12.4 qdel_cleanup

NAME

```
qdel_cleanup -- ???
```

SYNOPSIS

```
qdel_cleanup { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12.5 qdel_delete_job_0

NAME

```
qdel_delete_job_0 -- ???
```

SYNOPSIS

```
qdel_delete_job_0 { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12.6 qdel_delete_negative_jobid

NAME

```
qdel_delete_negative_jobid -- ???
```

SYNOPSIS

```
qdel_delete_negative_jobid { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12.7 qdel_delete_unkown_jobid

NAME

```
qdel_delete_unkown_jobid -- ???
```

SYNOPSIS

```
qdel_delete_unkown_jobid { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

12.8 qdel_force

NAME

```
qdel_force -- ???
```

SYNOPSIS

```
qdel_force { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

12.9 qdel_help

NAME

```
qdel_help -- ???
```

SYNOPSIS

```
qdel_help { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

12.10 qdel_job_task_list

NAME

```
qdel_job_task_list -- ???
```

SYNOPSIS

```
qdel_job_task_list { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

12.11 qdel_setup

NAME

```
qdel_setup -- ???
```

SYNOPSIS

```
qdel_setup { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12.12 qdel_uall

NAME

```
qdel_uall -- ???
```

SYNOPSIS

```
qdel_uall { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12.13 qdel_user_list

NAME

```
qdel_user_list -- ???
```

SYNOPSIS

```
qdel_user_list { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12.14 qdel_verify

NAME

```
qdel_verify -- ???
```

SYNOPSIS

```
qdel_verify { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

12.15 submit_testjobs

NAME

```
submit_testjobs -- ???
```

SYNOPSIS

```
submit_testjobs { { user "" } }
```

FUNCTION

```
???
```

INPUTS

```
{ user "" } - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

13 qmod

13.1 addqueue

NAME

```
addqueue -- ???
```

SYNOPSIS

```
addqueue { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

13.2 qmod_check_default_status

NAME

```
qmod_check_default_status -- ???
```

SYNOPSIS

```
qmod_check_default_status { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

13.3 qmod_clearerrorstate

NAME

```
qmod_clearerrorstate -- ???
```

SYNOPSIS

```
qmod_clearerrorstate { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

13.4 qmod_disable

NAME

```
qmod_disable -- ???
```

SYNOPSIS

```
qmod_disable { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

13.5 qmod_enable

NAME

```
qmod_enable -- ???
```

SYNOPSIS

```
qmod_enable { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

13.6 qmod_forceaction

NAME

```
qmod_forceaction -- ???
```

SYNOPSIS

```
qmod_forceaction { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

13.7 qmod_help

NAME

```
qmod_help -- ???
```

SYNOPSIS

```
qmod_help { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

13.8 qmod_suspend

NAME

```
qmod_suspend -- ???
```

SYNOPSIS

```
qmod_suspend { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

13.9 qmod_unsuspend

NAME

```
qmod_unsuspend -- ???
```

SYNOPSIS

```
qmod_unsuspend { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

13.10 qmod_verify

NAME

```
qmod_verify -- ???
```

SYNOPSIS

```
qmod_verify { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

13.11 removequeue

NAME

```
removequeue -- ???
```

SYNOPSIS

```
removequeue { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

14 qrsh

14.1 check_qsub_gid_output

NAME

```
check_qsub_gid_output -- ???
```

SYNOPSIS

```
check_qsub_gid_output { output check_group }
```

FUNCTION

```
???
```

INPUTS

```
output      - ???
check_group - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

14.2 init_level

NAME

```
init_level -- ???
```

SYNOPSIS

```
init_level { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

14.3 qrsh_accounting

NAME

```
qrsh_accounting -- ???
```

SYNOPSIS

```
qrsh_accounting { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

14.4 qrsh_alltoall

NAME

```
qrsh_alltoall -- ???
```

SYNOPSIS

```
qrsh_alltoall { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

14.5 qrsh_batch

NAME

```
qrsh_batch -- ???
```

SYNOPSIS

```
qrsh_batch { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

14.6 qrsh_delete

NAME

```
qrsh_delete -- ???
```

SYNOPSIS

```
qrsh_delete { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

14.7 qrsh_function

NAME

```
qrsh_function -- ???
```

SYNOPSIS

```
qrsh_function { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

14.8 qrsh_limits

NAME

```
qrsh_limits -- ???
```

SYNOPSIS

```
qrsh_limits { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

14.9 qrsh_qsub_gid

NAME

```
qrsh_qsub_gid -- ???
```

SYNOPSIS

```
qrsh_qsub_gid { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

14.10 qrsh_suspend

NAME

```
qrsh_suspend -- ???
```

SYNOPSIS

```
qrsh_suspend { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

14.11 qrsh_terminate

NAME

```
qrsh_terminate -- ???
```

SYNOPSIS

```
qrsh_terminate { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

14.12 qrsh_trap

NAME

```
qrsh_trap -- ???
```

SYNOPSIS

```
qrsh_trap { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

15 qstat

15.1 check_core_queues

NAME

`check_core_queues -- ???`

SYNOPSIS

`check_core_queues { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See '/'

15.2 get_numb_proc

NAME

`get_numb_proc -- ???`

SYNOPSIS

`get_numb_proc { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See '/'

16 qsub

16.1 check_deadline

NAME

`check_deadline -- ???`

SYNOPSIS

`check_deadline { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See '/'

16.2 check_hold

NAME

`check_hold -- ???`

SYNOPSIS

`check_hold { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See '/'

16.3 check_huge_script

NAME

```
check_huge_script -- ???
```

SYNOPSIS

```
check_huge_script { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.4 check_option_

NAME

```
check_option_@ -- ???
```

SYNOPSIS

```
check_option_@ { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.5 check_option_A

NAME

```
check_option_A -- ???
```

SYNOPSIS

```
check_option_A { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.6 check_option_C

NAME

```
check_option_C -- ???
```

SYNOPSIS

```
check_option_C { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.7 check_option_M

NAME

```
check_option_M -- ???
```

SYNOPSIS

```
check_option_M { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.8 check_option_N

NAME

```
check_option_N -- ???
```

SYNOPSIS

```
check_option_N { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.9 check_option_P

NAME

```
check_option_P -- ???
```

SYNOPSIS

```
check_option_P { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.10 check_option_S

NAME

```
check_option_S -- ???
```

SYNOPSIS

```
check_option_S { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.11 check_option_V

NAME

```
check_option_V -- ???
```

SYNOPSIS

```
check_option_V { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

16.12 check_option_ac

NAME

```
check_option_ac -- ???
```

SYNOPSIS

```
check_option_ac { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

16.13 check_option_c

NAME

```
check_option_c -- ???
```

SYNOPSIS

```
check_option_c { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.14 check_option_ckpt

NAME

```
check_option_ckpt -- ???
```

SYNOPSIS

```
check_option_ckpt { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.15 check_option_clear

NAME

```
check_option_clear -- ???
```

SYNOPSIS

```
check_option_clear { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

16.16 check_option_cwd

NAME

```
check_option_cwd -- ???
```

SYNOPSIS

```
check_option_cwd { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

16.17 check_option_dc

NAME

```
check_option_dc -- ???
```

SYNOPSIS

```
check_option_dc { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.18 check_option_e

NAME

```
check_option_e -- ???
```

SYNOPSIS

```
check_option_e { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.19 check_option_hard

NAME

```
check_option_hard -- ???
```

SYNOPSIS

```
check_option_hard { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.20 check_option_help

NAME

```
check_option_help -- ???
```

SYNOPSIS

```
check_option_help { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.21 check_option_hold_jid

NAME

```
check_option_hold_jid -- ???
```

SYNOPSIS

```
check_option_hold_jid { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.22 check_option_j_n

NAME

```
check_option_j_n -- ???
```

SYNOPSIS

```
check_option_j_n { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.23 check_option_j_y

NAME

```
check_option_j_y -- ???
```

SYNOPSIS

```
check_option_j_y { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.24 check_option_l

NAME

```
check_option_l -- ???
```

SYNOPSIS

```
check_option_l { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.25 check_option_m

NAME

```
check_option_m -- ???
```

SYNOPSIS

```
check_option_m { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.26 check_option_notify

NAME

```
check_option_notify -- ???
```

SYNOPSIS

```
check_option_notify { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.27 check_option_now_no

NAME

`check_option_now_no -- ???`

SYNOPSIS

`check_option_now_no { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See '/'

16.28 check_option_now_yes

NAME

`check_option_now_yes -- ???`

SYNOPSIS

`check_option_now_yes { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See '/'

16.29 check_option_o

NAME

```
check_option_o -- ???
```

SYNOPSIS

```
check_option_o { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.30 check_option_p

NAME

```
check_option_p -- ???
```

SYNOPSIS

```
check_option_p { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.31 check_option_pe

NAME

```
check_option_pe -- ???
```

SYNOPSIS

```
check_option_pe { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

16.32 check_option_q

NAME

```
check_option_q -- ???
```

SYNOPSIS

```
check_option_q { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

16.33 check_option_qs_args

NAME

```
check_option_qs_args -- ???
```

SYNOPSIS

```
check_option_qs_args { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.34 check_option_r_n

NAME

```
check_option_r_n -- ???
```

SYNOPSIS

```
check_option_r_n { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.35 check_option_r_y

NAME

```
check_option_r_y -- ???
```

SYNOPSIS

```
check_option_r_y { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.36 check_option_sc

NAME

```
check_option_sc -- ???
```

SYNOPSIS

```
check_option_sc { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.37 check_option_soft

NAME

```
check_option_soft -- ???
```

SYNOPSIS

```
check_option_soft { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.38 check_option_t

NAME

```
check_option_t -- ???
```

SYNOPSIS

```
check_option_t { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.39 check_option_v

NAME

```
check_option_v -- ???
```

SYNOPSIS

```
check_option_v { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.40 check_option_verify

NAME

```
check_option_verify -- ???
```

SYNOPSIS

```
check_option_verify { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.41 check_option_w

NAME

```
check_option_w -- ???
```

SYNOPSIS

```
check_option_w { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.42 check_start_time

NAME

```
check_start_time -- ???
```

SYNOPSIS

```
check_start_time { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.43 check_submit

NAME

```
check_submit -- ???
```

SYNOPSIS

```
check_submit { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.44 select_queue

NAME

```
select_queue -- ???
```

SYNOPSIS

```
select_queue { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

16.45 setup_output_directory

NAME

```
setup_output_directory -- ???
```

SYNOPSIS

```
setup_output_directory { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

17 remote_procedures

17.1 close_spawn_process

NAME

```
close_spawn_process -- close open spawn process id
```

SYNOPSIS

```
close_spawn_process { id }
```

FUNCTION

```
This procedure will close the process associated with the spawn id
returned from the procedures open_spawn_process or open_root_spawn_process
```

INPUTS

```
id - spawn process id (returned from open_spawn_process or
     open_root_spawn_process)
```

RESULT

```
exit state of the "spawned" process
```

EXAMPLE

```
see open_root_spawn_process or open_spawn_process
```

NOTES

```
After a process is "spawned" with the  open_spawn_process procedure it
must be closed with the close_spawn_process procedure. id is the return
value of open_spawn_process or open_root_spawn_process.
If a open spawn process id is not closed, it will not free the file
descriptor for that id. If all file descriptors are used, no new spawn
process can be forked!
```

SEE ALSO

See Section 17.4 [remote_procedures open_spawn_process], page 163.
See Section 17.3 [remote_procedures open_root_spawn_process], page 162.
See Section 17.1 [remote_procedures close_spawn_process], page 161.
See Section 17.5 [remote_procedures run_command_as_user], page 164.
See Section 17.7 [remote_procedures start_remote_tcl_prog], page 166.
See Section 17.6 [remote_procedures start_remote_prog], page 165.

17.2 open_remote_spawn_process

NAME

```
open_remote_spawn_process -- ???
```

SYNOPSIS

```
open_remote_spawn_process { hostname user exec_command exec_arguments { bac
```

FUNCTION

```
???
```

INPUTS

```
hostname         - ???
user             - ???
exec_command     - ???
exec_arguments   - ???
{ background 0 } - if not 0 -> start command with "&" in background
```

RESULT

```
???
```

EXAMPLE

```
set id [open_remote_spawn_process "boromir" "testuser" "ls" "-la"]
set do_stop 0
set output ""
while { $do_stop == 0 } {
   expect {
      timeout { set do_stop 1 }
      eof { set do_stop 1 }
      "*\r" {
         set output "$output$expect_out(0,string)"
      }
   }
}
close_spawn_process $id
puts $CHECK_OUTPUT ">>> output start <<<"
puts $CHECK_OUTPUT $output
puts $CHECK_OUTPUT ">>> output end <<<"
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

17.3 open_root_spawn_process

NAME

```
open_root_spawn_process -- starrrt process as root with spawn command
```

SYNOPSIS

```
open_root_spawn_process { args }
```

FUNCTION

```
Starts process given in "args" as user "root" and returns its spawn id
and pid in a list. The root password is sent when the su command is
asking for the root password.
The first list element is the pid and the second is the spawn id. The
return value is used in close_spawn_process to close the connection to
this process.
```

INPUTS

```
args - full argument list of the process to start
```

RESULT

```
tcl list with id and pid of the process

- first element is the pid
- second element is the spawn id
```

EXAMPLE

```
set id [
  open_spawn_process "id"
]
set timeout 60
expect {
  timeout { puts "timeout" }
  "root" { puts "we have root access" }
}
puts "pid: [ lindex $id 0]"
puts "spawn id: [ lindex $id 1]"
close_spawn_process $id
```

SEE ALSO

See Section 17.4 [remote_procedures open_spawn_process], page 163.
See Section 17.3 [remote_procedures open_root_spawn_process], page 162.
See Section 17.1 [remote_procedures close_spawn_process], page 161.
See Section 17.5 [remote_procedures run_command_as_user], page 164.
See Section 17.7 [remote_procedures start_remote_tcl_prog], page 166.
See Section 17.6 [remote_procedures start_remote_prog], page 165.

17.4 open_spawn_process

NAME

```
open_spawn_process -- start process with the expect "spawn" command
```

SYNOPSIS

```
open_spawn_process { args }
```

FUNCTION

```
Starts process given in "args" and returns its spawn id and pid in a list.
The first list element is the pid and the second is the spawn id. The retu
value is used in close_spawn_process to close the connection to this
process.
```

INPUTS

```
args - full argument list of the process to start
```

RESULT

```
tcl list with id and pid of the process

- first element is the pid
- second element is the spawn id
```

EXAMPLE

```
set id [
  open_spawn_process "$CHECK_PRODUCT_ROOT/bin/$CHECK_ARCH/qconf" "-dq" "$q
]
expect {
  ...
}
puts "pid: [ lindex $id 0]"
puts "spawn id: [ lindex $id 1]"
close_spawn_process $id
```

NOTES

```
always close an opened spawn id with the procedure close_spawn_process
```

SEE ALSO

See Section 17.4 [remote_procedures open_spawn_process], page 163.
See Section 17.3 [remote_procedures open_root_spawn_process], page 162.
See Section 17.1 [remote_procedures close_spawn_process], page 161.
See Section 17.5 [remote_procedures run_command_as_user], page 164.
See Section 17.7 [remote_procedures start_remote_tcl_prog], page 166.
See Section 17.6 [remote_procedures start_remote_prog], page 165.

17.5 run_command_as_user

NAME

```
run_command_as_user -- start proccess under a specific user account
```

SYNOPSIS

```
run_command_as_user { hostname user command args counter }
```

FUNCTION

```
This procedure is using start_remote_prog to start a binary or a skript
file under a specific user account.
```

INPUTS

```
hostname - host where the command should be started
user     - system user name who should start the command
command  - command name (if no full path is given, the product
           root path will be used)
args     - command arguments
counter  - run the command $counter times
```

RESULT

```
the command output
```

EXAMPLE

```
set jobargs "/home/me/testjob.sh"
set result [ run_command_as_user "expo1" "user1" "qsub" "$jobargs" 5]
puts $result
```

NOTES

```
This procedure starts the script file remote_submit.sh in the scripts
directory of the testsuite. This script is sourcing the
default/common/settings.sh file of the cluster. If the command parameter
has no full path entry it will add the $CHECK_PRODUCT_ROOT path in
front of the command.
```

SEE ALSO

See Section 17.4 [remote_procedures open_spawn_process], page 163.
See Section 17.3 [remote_procedures open_root_spawn_process], page 162.
See Section 17.1 [remote_procedures close_spawn_process], page 161.
See Section 17.5 [remote_procedures run_command_as_user], page 164.
See Section 17.7 [remote_procedures start_remote_tcl_prog], page 166.
See Section 17.6 [remote_procedures start_remote_prog], page 165.

17.6 start_remote_prog

NAME

```
start_remote_prog() -- ???
```

SYNOPSIS

```
start_remote_prog { hostname user exec_command exec_arguments
{exit_var prg_exit_state} {mytimeout 60} {background 0} }
```

FUNCTION

```
???
```

INPUTS

```
hostname                   - ???
user                       - ???
exec_command               - ???
exec_arguments             - ???
{exit_var prg_exit_state}  - ???
{mytimeout 60}             - ???
{background 0}             - if not 0 -> start remote prog in background
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

17.7 start_remote_tcl_prog

NAME

```
start_remote_tcl_prog -- ???
```

SYNOPSIS

```
start_remote_tcl_prog { host user tcl_file tcl_procedure tcl_procargs }
```

FUNCTION

```
???
```

INPUTS

```
host          - ???
user          - ???
tcl_file      - ???
tcl_procedure - ???
tcl_procargs  - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

17.8 test

NAME

```
test -- ???
```

SYNOPSIS

```
test { m p }
```

FUNCTION

```
???
```

INPUTS

```
m - ???
p - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

`???`

BUGS

`???`

SEE ALSO

See ‘`/`’

18 sge_procedures

18.1 add_calendar

NAME

```
add_calendar -- add new calendar definition object
```

SYNOPSIS

```
add_calendar { change_array }
```

FUNCTION

```
This procedure will add/define a new calendar definition object
```

INPUTS

```
change_array - name of an array variable that will be set by add_calendar
```

RESULT

```
-1   timeout error
-2   callendar allready exists
 0   ok
```

EXAMPLE

```
set new_cal(calendar_name)  "always_suspend"
set new_cal(year)           "NONE"
set new_cal(week)           "mon-sun=0-24=suspended"
```

NOTES

```
The array should look like this:

set change_array(calendar_name) "mycalendar"
set change_array(year)            "NONE"
set change-array(week)          "mon-sun=0-24=suspended"
....
(every value that is set will be changed)

Here the possible change_array values with some typical settings:

attribute(calendar_name) "test"
attribute(year)          "NONE"
attribute(week)          "NONE"
```

SEE ALSO

See ‘/’

18.2 add_checkpointobj

NAME

```
add_checkpointobj -- add a new checkpoint definiton object
```

SYNOPSIS

```
add_checkpointobj { change_array }
```

FUNCTION

```
This procedure will add a new checkpoint definition object
```

INPUTS

```
change_array - name of an array variable that will be set by
               add_checkpointobj
```

NOTES

```
The array should look like follows:

set myarray(ckpt_name) "myname"
set myarray(queue_list) "big.q"
...

Here the possbile change_array values with some typical settings:

ckpt_name          test
interface          userdefined
ckpt_command       none
migr_command       none
restart_command    none
clean_command      none
ckpt_dir           /tmp
queue_list         NONE
signal             none
when               sx
```

RESULT

```
 0   - ok
-1   - timeout error
-2   - object already exists
-3   - queue reference does not exist
```

SEE ALSO

See Section 18.8 [sge_procedures del_checkpointobj], page 174.

18.3 add_pe

NAME

```
add_pe -- add new parallel environment definition object
```

SYNOPSIS

```
add_pe { change_array }
```

FUNCTION

```
This procedure will create a new pe (parallel environemnt) definition
object.
```

INPUTS

```
change_array - name of an array variable that will be set by add_pe
```

RESULT

```
 0 - ok
-1 - timeout error
-2 - pe already exists
-3 - could not add pe
```

EXAMPLE

```
set mype(pe_name) "mype"
set mype(user_list) "user1"
add_pe pe_name
```

NOTES

```
The array should look like this:

set change_array(pe_name)          "mype"
set change_array(user_list)        "crei"
....
(every value that is set will be changed)

Here the possible change_array values with some typical settings:

pe_name            testpe
queue_list         NONE
slots              0
user_lists         NONE
xuser_lists        NONE
start_proc_args    /bin/true
stop_proc_args     /bin/true
allocation_rule    $pe_slots
control_slaves     FALSE
job_is_first_task  TRUE
```

SEE ALSO

See Section 18.9 [sge_procedures del_pe], page 174.

18.4 add_prj

NAME

```
add_prj -- ???
```

SYNOPSIS

```
add_prj { change_array }
```

FUNCTION

```
???
```

INPUTS

```
change_array - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.5 add_queue

NAME

```
add_queue -- Add a new queue configuration object
```

SYNOPSIS

```
add_queue { change_array {fast_add 0} }
```

FUNCTION

```
Add a new queue configuration object corresponding to the content of
the change_array.
```

INPUTS

```
change_array - name of an array variable that will be set by get_config
{fast_add 0} - if not 0 the add_queue procedure will use a file for
               queue configuration. (faster) (qconf -Aq, not qconf -aq)
```

RESULT

```
-1   timeout error
-2   queue allready exists
 0   ok
```

EXAMPLE

```
set new_queue(qname)    "new.q"
set new_queue(hostname) "expo1"
add_queue new_queue
```

NOTES

```
the array should look like this:

set change_array(qname) MYHOST
set change_array(hostname) MYHOST.domain
....
(every value that is set will be changed)

here is a list of all guilty array names (template queue):

change_array(qname)                 "template"
change_array(hostname)              "unknown"
change_array(seq_no)                "0"
change_array(load_thresholds)       "np_load_avg=1.75"
change_array(suspend_thresholds)    "NONE"
change_array(nsuspend)              "0"
change_array(suspend_interval)      "00:05:00"
change_array(priority)              "0"
change_array(max_migr_time)         "0"
change_array(migr_load_thresholds) "np_load_avg=5.00"
change_array(max_no_migr)           "00:02:00"
change_array(min_cpu_interval)      "00:05:00"
change_array(processors)            "UNDEFINED"
change_array(qtype)                 "BATCH INTERACTIVE"
change_array(rerun)                 "FALSE"
change_array(slots)                 "1"
change_array(tmpdir)                "/tmp"
change_array(shell)                 "/bin/csh"
change_array(shell_start_mode)      "NONE"
change_array(klog)                  "/usr/local/bin/klog"
change_array(prolog)                "NONE"
change_array(epilog)                "NONE"
change_array(starter_method)        "NONE"
change_array(suspend_method)        "NONE"
change_array(resume_method)         "NONE"
change_array(terminate_method)      "NONE"
change_array(reauth_time)           "01:40:00"
change_array(notify)                "00:00:60"
change_array(owner_list)            "NONE"
change_array(user_lists)            "NONE"
change_array(xuser_lists)           "NONE"
change_array(subordinate_list)      "NONE"
change_array(complex_list)          "NONE"
change_array(complex_values)        "NONE"
change_array(projects)              "NONE"
change_array(xprojects)             "NONE"
change_array(calendar)              "NONE"
change_array(initial_state)         "default"
change_array(fshare)                "0"
change_array(oticket)               "0"
change_array(s_rt)                  "INFINITY"
change_array(h_rt)                  "INFINITY"
change_array(s_cpu)                 "INFINITY"
change_array(h_cpu)                 "INFINITY"
change_array(s_fsize)               "INFINITY"
change_array(h_fsize)               "INFINITY"
change_array(s_data)                "INFINITY"
change_array(h_data)                "INFINITY"
change_array(s_stack)               "INFINITY"
change_array(h_stack)               "INFINITY"
change_array(s_core)                "INFINITY"
change_array(h_core)                "INFINITY"
change_array(s_rss)                 "INFINITY"
change_array(h_rss)                 "INFINITY"
change_array(s_vmem)                "INFINITY"
change_array(h_vmem)                "INFINITY"
```

SEE ALSO

See Section 18.38 [sge_procedures mqattr], page 193.
See Section 18.47 [sge_procedures set_queue], page 200.
See Section 18.5 [sge_procedures add_queue], page 171.
See Section 18.11 [sge_procedures del_queue], page 175.
See Section 18.26 [sge_procedures get_queue], page 185.
See Section 18.59 [sge_procedures suspend_queue], page 209.
See Section 18.62 [sge_procedures unsuspend_queue], page 211.
See Section 18.13 [sge_procedures disable_queue], page 176.
See Section 18.14 [sge_procedures enable_queue], page 177.

18.6 are_master_and_scheduler_running

NAME

```
are_master_and_scheduler_running -- ???
```

SYNOPSIS

```
are_master_and_scheduler_running { hostname qmaster_spool_dir }
```

FUNCTION

```
???
```

INPUTS

```
hostname          - ???
qmaster_spool_dir - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.7 del_calendar

NAME

```
del_calendar -- ???
```

SYNOPSIS

```
del_calendar { mycal_name }
```

FUNCTION

```
???
```

INPUTS

```
mycal_name - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.8 del_checkpointobj

NAME

```
del_checkpointobj -- delete checkpoint object definition
```

SYNOPSIS

```
del_checkpointobj { checkpoint_name }
```

FUNCTION

```
This procedure will delete a checkpoint object definition by its name.
```

INPUTS

```
checkpoint_name - name of the checkpoint object
```

RESULT

```
 0   - ok
-1   - timeout error
```

SEE ALSO

See Section 18.2 [sge_procedures add_checkpointobj], page 168.

18.9 del_pe

NAME

```
del_pe -- delete parallel environment object definition
```

SYNOPSIS

```
del_pe { mype_name }
```

FUNCTION

```
This procedure will delete a existing parallel environment, defined with
sge_procedures/add_pe.
```

INPUTS

```
mype_name - name of parallel environment to delete
```

RESULT

```
0  - ok
-1 - timeout error
```

SEE ALSO

See Section 18.3 [sge_procedures add_pe], page 169.

18.10 del_prj

NAME

```
del_prj -- ???
```

SYNOPSIS

```
del_prj { myprj_name }
```

FUNCTION

```
???
```

INPUTS

```
myprj_name - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.11 del_queue

NAME

```
del_queue -- delete a queue
```

SYNOPSIS

```
del_queue { q_name }
```

FUNCTION

```
remove a queue from the qmaster configuration
```

INPUTS

```
q_name - name of the queue to delete
```

RESULT

```
0  : ok
-1 : timeout error
```

EXAMPLE

```
del_queue "my_own_queue.q"
```

SEE ALSO

See Section 18.38 [sge_procedures mqattr], page 193.
See Section 18.47 [sge_procedures set_queue], page 200.
See Section 18.5 [sge_procedures add_queue], page 171.
See Section 18.11 [sge_procedures del_queue], page 175.
See Section 18.26 [sge_procedures get_queue], page 185.
See Section 18.59 [sge_procedures suspend_queue], page 209.
See Section 18.62 [sge_procedures unsuspend_queue], page 211.
See Section 18.13 [sge_procedures disable_queue], page 176.
See Section 18.14 [sge_procedures enable_queue], page 177.

18.12 delete_job

NAME

```
delete_job -- delete job with jobid
```

SYNOPSIS

```
delete_job { jobid }
```

FUNCTION

```
This procedure will delete the job with the given jobid
```

INPUTS

```
jobid - job identification number
```

RESULT

```
 0    - ok
-1    - timeout error
```

SEE ALSO

See Section 18.57 [sge_procedures submit_job], page 208.

18.13 disable_queue

NAME

```
disable_queue -- disable queues
```

SYNOPSIS

```
disable_queue { queue }
```

FUNCTION

```
Disable the given queue/queue list
```

INPUTS

```
queue - name of queues to disable
```

RESULT

```
 0  - ok
-1  - error
```

SEE ALSO

See Section 18.38 [sge_procedures mqattr], page 193.
See Section 18.47 [sge_procedures set_queue], page 200.
See Section 18.5 [sge_procedures add_queue], page 171.
See Section 18.11 [sge_procedures del_queue], page 175.
See Section 18.26 [sge_procedures get_queue], page 185.
See Section 18.59 [sge_procedures suspend_queue], page 209.
See Section 18.62 [sge_procedures unsuspend_queue], page 211.
See Section 18.13 [sge_procedures disable_queue], page 176.
See Section 18.14 [sge_procedures enable_queue], page 177.

18.14 enable_queue

NAME

```
enable_queue -- enable queuelist
```

SYNOPSIS

```
enable_queue { queue }
```

FUNCTION

```
This procedure enables a given queuelist by calling the qmod -e binary
```

INPUTS

```
queue - name of queues to enable (list)
```

RESULT

```
 0  - ok
-1  - on error
```

SEE ALSO

See Section 18.38 [sge_procedures mqattr], page 193.
See Section 18.47 [sge_procedures set_queue], page 200.
See Section 18.5 [sge_procedures add_queue], page 171.
See Section 18.11 [sge_procedures del_queue], page 175.
See Section 18.26 [sge_procedures get_queue], page 185.
See Section 18.59 [sge_procedures suspend_queue], page 209.
See Section 18.62 [sge_procedures unsuspend_queue], page 211.
See Section 18.13 [sge_procedures disable_queue], page 176.
See Section 18.14 [sge_procedures enable_queue], page 177.

18.15 get_config

NAME

```
get_config -- get global or host configuration settings
```

SYNOPSIS

```
get_config { change_array {host "global"} }
```

FUNCTION

```
Get the global or host specific configuration settings.
```

INPUTS

```
change_array    - name of an array variable that will get set by
                  get_config
{host "global"} - get configuration for a specific hostname (host)
                  or get the global configuration (global)
```

RESULT

```
The change_array variable is build as follows:

set change_array(xterm)   "/bin/xterm"
set change_array(enforce_project) "true"
...
```

EXAMPLE

```
get_config gcluster1 lobal
puts $cluster1(qmaster_spool_dir)

Here the possible change_array values with some typical settings:

qmaster_spool_dir    /../default/spool/qmaster
execd_spool_dir      /../default/spool
qsi_common_dir       /../default/common/qsi
binary_path          /../bin
mailer               /usr/sbin/Mail
xterm                /usr/bin/X11/xterm
load_sensor          none
prolog               none
epilog               none
shell_start_mode     posix_compliant
login_shells         sh,ksh,csh,tcsh
min_uid              0
min_gid              0
user_lists           none
xuser_lists          none
projects             none
xprojects            none
load_report_time     00:01:00
stat_log_time        12:00:00
max_unheard          00:02:30
loglevel             log_info
enforce_project      false
administrator_mail   none
set_token_cmd        none
pag_cmd              none
token_extend_time    none
shepherd_cmd         none
qmaster_params       none
schedd_params        none
execd_params         none
```

```
finished_jobs        0
gid_range            13001-13100
admin_user           crei
qlogin_command       telnet
qlogin_daemon        /usr/etc/telnetd
```

SEE ALSO

See Section 18.45 [sge_procedures set_config], page 197.

18.16 get_execd_spool_dir

NAME

```
get_execd_spool_dir() -- return spool dir for exec host
```

SYNOPSIS

```
get_execd_spool_dir { host }
```

FUNCTION

```
This procedure returns the actual execd spool directory on the given host.
If no local spool directory is specified for this host, the global
configuration is used. If an error accurs the procedure returns "".
```

INPUTS

```
host - host name with execd installed on
```

RESULT

```
string
```

SEE ALSO

See Section 18.25 [sge_procedures get_qmaster_spool_dir], page 184.

18.17 get_exechost

NAME

```
get_exechost -- get exec host configuration
```

SYNOPSIS

```
get_exechost { change_array host }
```

FUNCTION

```
Get the exec host specific configuration settings. The given variable
is used to save the configuration settings.
```

INPUTS

```
change_array - name of an array variable that will get set by get_exechost
host         - name of an execution host
```

RESULT

```
The array is build like follows:

set change_array(user_list)   "deadlineusers"
set change_array(load_scaling) "NONE"
```

```
....

Here the possible change_array values with some typical settings:

hostname                     myhost.mydomain
load_scaling                 NONE
complex_list                 test
complex_values               NONE
user_lists                   deadlineusers
xuser_lists                  NONE
projects                     NONE
xprojects                    NONE
usage_scaling                NONE
resource_capability_factor 0.000000
```

EXAMPLE

```
get_exechost change_array expo1
puts $change_array(user_list)
```

SEE ALSO

See Section 18.46 [sge_procedures set_exechost], page 199.

18.18 get_extended_job_info

NAME

```
get_extended_job_info -- get extended job information (qstat ..)
```

SYNOPSIS

```
get_extended_job_info { jobid {variable job_info} }
```

FUNCTION

```
This procedure is calling the qstat (qstat -ext if sgeee) and returns
the output of the qstat in array form.
```

INPUTS

```
jobid               - job identifaction number
{variable job_info} - name of variable array to store the output
```

RESULT

```
0, if job was not found
1, if job was found

fills array $variable with info found in qstat output with the following sy
id
prior
name
user
state
time (submit or starttime) [UNIX-timestamp]
queue
master
```

```
jatask

additional entries in case of SGEEE system:
project
department
deadline [UNIX-timestamp]
cpu [s]
mem [GBs]
io [?]
tckts
ovrts
otckt
dtckt
ftckt
stckt
share
```

EXAMPLE

```
proc testproc ... {
   ...
   if {[get_extended_job_info $job_id] } {
      if { $job_info(cpu) < 10 } {
         add_proc_error "testproc" -1 "online usage probably does not work
      }
   } else {
      add_proc_error "testproc" -1 "get_extended_jobinfo failed for job $jo
   }
   ...
   set_error 0 "ok"
}
```

SEE ALSO

See Section 18.22 [sge_procedures get_job_info], page 183.
See Section 18.29 [sge_procedures get_standard_job_info], page 188.
See Section 18.18 [sge_procedures get_extended_job_info], page 180.

18.19 get_gid_range

NAME

```
get_gid_range() -- get gid range for user
```

SYNOPSIS

```
get_gid_range { user port }
```

FUNCTION

```
This procedure ist used in the install_core_system test. It returns the
gid range of the requested user and port
```

INPUTS

```
user - user name
port - port number on which the cluster commd is running
```

RESULT

```
gid range, e.g. 13501-13700
```

SEE ALSO

See '/'

18.20 get_grppid_of_job

NAME

```
get_grppid_of_job -- get grppid of job
```

SYNOPSIS

```
get_grppid_of_job { jobid }
```

FUNCTION

```
This procedure opens the job_pid file in the execution host spool directory
and returns the content of this file (grppid).
```

INPUTS

```
jobid - identification number of job
```

RESULT

```
grppid of job
```

SEE ALSO

See Section 18.30 [sge_procedures get_suspend_state_of_job], page 189.

18.21 get_hosts

NAME

```
get_hosts -- ???
```

SYNOPSIS

```
get_hosts { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.22 get_job_info

NAME

```
get_job_info -- get qstat -ext jobinformation
```

SYNOPSIS

```
get_job_info { jobid }
```

FUNCTION

```
This procedure runs the qstat -ext command and returns the output
```

INPUTS

```
jobid - job id (if job id = -1 the complete joblist is returned)
```

RESULT

```
"" if job was not found or the call fails
output of qstat -ext
```

SEE ALSO

See Section 18.22 [sge_procedures get_job_info], page 183.
See Section 18.29 [sge_procedures get_standard_job_info], page 188.
See Section 18.18 [sge_procedures get_extended_job_info], page 180.

18.23 get_loadsensor_path

NAME

```
get_loadsensor_path -- ???
```

SYNOPSIS

```
get_loadsensor_path { arch }
```

FUNCTION

```
???
```

INPUTS

```
arch - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.24 get_qacct

NAME

```
get_qacct -- get job accounting information
```

SYNOPSIS

```
get_qacct { jobid {variable qacct_info} }
```

FUNCTION

```
This procedure will parse the qacct output for the given job id and fill
up the given variable name with information.
```

INPUTS

```
jobid                 - job identification number
{variable qacct_info} - name of variable to save the results
```

RESULT

```
0, if job was not found
1, if job was found
```

EXAMPLE

```
if { [get_qacct $job_id] == 0 } {
   set_error -1 "qacct for job $job_id on host $host failed"
} else {
   set cpu [expr $qacct_info(ru_utime) + $qacct_info(ru_stime)]
   if { $cpu < 30 } {
      set_error -1 "cpu entry in accounting ($qacct_info(cpu)) seems
                    to be wrong for job $job_id on host $host"
   }

   if { $CHECK_PRODUCT_TYPE == "sgeee" } {
      # compute absolute diffence between cpu and ru_utime + ru_stime
      set difference [expr $cpu - $qacct_info(cpu)]
      set difference [expr $difference * $difference]
      if { $difference > 1 } {
         set_error -1 "accounting: cpu($qacct_info(cpu)) is not the
                       sum of ru_utime and ru_stime ($cpu) for
                       job $job_id on host $host"
      }
   }
}
```

NOTES

```
look at parser/parse_qacct for more information
```

SEE ALSO

See Section 8.7 [parser parse_qacct], page 76.

18.25 get_qmaster_spool_dir

NAME

```
get_qmaster_spool_dir() -- return path to qmaster spool directory
```

SYNOPSIS

```
get_qmaster_spool_dir { }
```

FUNCTION

```
This procedure returns the actual qmaster spool directory
(or "" in case of an error)
```

RESULT

```
string with actual spool directory of qmaster
```

SEE ALSO

See Section 18.16 [sge_procedures get_execd_spool_dir], page 179.

18.26 get_queue

NAME

```
get_queue -- get queue configuration information
```

SYNOPSIS

```
get_queue { q_name change_array }
```

FUNCTION

```
Get the actual configuration settings for the named queue
```

INPUTS

```
q_name       - name of the queue
change_array - name of an array variable that will get set by get_config
```

EXAMPLE

```
get_queue "myqueue.q" qinfo
puts qinfo(seq_no)
```

NOTES

```
the array should look like this:

set change_array(qname) MYHOST
set change_array(hostname) MYHOST.domain
....
(every value that is set will be changed)

here is a list of all guilty array names (template queue):

change_array(qname)                "template"
change_array(hostname)             "unknown"
change_array(seq_no)               "0"
change_array(load_thresholds)      "np_load_avg=1.75"
change_array(suspend_thresholds)   "NONE"
change_array(nsuspend)             "0"
change_array(suspend_interval)     "00:05:00"
change_array(priority)             "0"
change_array(max_migr_time)        "0"
change_array(migr_load_thresholds) "np_load_avg=5.00"
change_array(max_no_migr)          "00:02:00"
change_array(min_cpu_interval)     "00:05:00"
change_array(processors)           "UNDEFINED"
change_array(qtype)                "BATCH INTERACTIVE"
change_array(rerun)                "FALSE"
change_array(slots)                "1"
change_array(tmpdir)               "/tmp"
change_array(shell)                "/bin/csh"
change_array(shell_start_mode)     "NONE"
change_array(klog)                 "/usr/local/bin/klog"
change_array(prolog)               "NONE"
change_array(epilog)               "NONE"
change_array(starter_method)       "NONE"
change_array(suspend_method)       "NONE"
change_array(resume_method)        "NONE"
change_array(terminate_method)     "NONE"
change_array(reauth_time)          "01:40:00"
change_array(notify)               "00:00:60"
change_array(owner_list)           "NONE"
change_array(user_lists)           "NONE"
change_array(xuser_lists)          "NONE"
change_array(subordinate_list)     "NONE"
change_array(complex_list)         "NONE"
change_array(complex_values)       "NONE"
change_array(projects)             "NONE"
change_array(xprojects)            "NONE"
change_array(calendar)             "NONE"
change_array(initial_state)        "default"
change_array(fshare)               "0"
change_array(oticket)              "0"
change_array(s_rt)                 "INFINITY"
change_array(h_rt)                 "INFINITY"
change_array(s_cpu)                "INFINITY"
change_array(h_cpu)                "INFINITY"
change_array(s_fsize)              "INFINITY"
change_array(h_fsize)              "INFINITY"
change_array(s_data)               "INFINITY"
change_array(h_data)               "INFINITY"
change_array(s_stack)              "INFINITY"
change_array(h_stack)              "INFINITY"
change_array(s_core)               "INFINITY"
change_array(h_core)               "INFINITY"
change_array(s_rss)                "INFINITY"
change_array(h_rss)                "INFINITY"
change_array(s_vmem)               "INFINITY"
change_array(h_vmem)               "INFINITY"
```

SEE ALSO

See Section 18.38 [sge_procedures mqattr], page 193.
See Section 18.47 [sge_procedures set_queue], page 200.
See Section 18.5 [sge_procedures add_queue], page 171.
See Section 18.11 [sge_procedures del_queue], page 175.
See Section 18.26 [sge_procedures get_queue], page 185.

See Section 18.59 [sge_procedures suspend_queue], page 209.
See Section 18.62 [sge_procedures unsuspend_queue], page 211.
See Section 18.13 [sge_procedures disable_queue], page 176.
See Section 18.14 [sge_procedures enable_queue], page 177.

18.27 get_queue_state

NAME

```
get_queue_state -- get the state of a queue
```

SYNOPSIS

```
get_queue_state { queue }
```

FUNCTION

```
This procedure returns the state of the queue by parsing output of qstat -
```

INPUTS

```
queue - name of the queue
```

RESULT

```
The return value can contain more than one state. Here is a list of possib
states:

u(nknown)
a(larm)
A(larm)
C(alendar  suspended)
s(uspended)
S(ubordinate)
d(isabled)
D(isabled)
E(rror)
```

18.28 get_schedd_config

NAME

```
get_schedd_config -- get scheduler configuration
```

SYNOPSIS

```
get_schedd_config { change_array }
```

FUNCTION

```
Get the current scheduler configuration
```

INPUTS

```
change_array - name of an array variable that will get set by
               get_schedd_config
```

EXAMPLE

```
get_schedd_config test
puts $test(schedule_interval)
```

NOTES

```
The array is build like follows:

set change_array(algorithm) default
set change_array(schedule_interval) 0:0:15
....

Here the possible change_array values with some typical settings:

algorithm                    "default"
schedule_interval            "0:0:15"
maxujobs                     "0"
maxgjobs                     "0"
queue_sort_method            "share"
user_sort                    "false"
job_load_adjustments         "np_load_avg=0.50"
load_adjustment_decay_time   "0:7:30"
load_formula                 "np_load_avg"
schedd_job_info              "true"


In case of a SGEEE - System:

sgeee_schedule_interval        "00:01:00"
halftime                     "0"
usage_weight_list            "cpu=0.34,mem=0.33,io=0.33"
compensation_factor          "5"
weight_user                  "0"
weight_project               "0"
weight_jobclass              "0"
weight_department            "0"
weight_job                   "0"
weight_tickets_functional    "0"
weight_tickets_share         "0"
weight_tickets_deadline      "10000"
```

SEE ALSO

See Section 18.48 [sge_procedures set_schedd_config], page 202.

18.29 get_standard_job_info

NAME

```
get_standard_job_info -- get jobinfo with qstat
```

SYNOPSIS

```
get_standard_job_info { jobid { add_empty 0} { get_all 0 } }
```

FUNCTION

```
This procedure will call the qstat command without arguments.
```

INPUTS

```
jobid           - job id
{ add_empty 0 } - if 1: add lines with does not contain a job id
                  information (SLAVE jobs)
{ get_all   0 } - if 1: get every output line (ignore job id)
```

RESULT

```
- info of qstat for jobid
- nothing if job was not found

each list element has following sublists:
job-ID        (index 0)
prior         (index 1)
name          (index 2)
user          (index 3)
state         (index 4)
submit/start  (index 5)
at            (index 6)
queue         (index 7)
master        (index 8)
ja-task-ID    (index 9)
```

EXAMPLE

```
set result [get_standard_job_info 5]
if { llength $results > 0 } {
   puts "user [lindex $result 3] submitted job 5"
}
```

SEE ALSO

See Section 18.22 [sge_procedures get_job_info], page 183.

See Section 18.29 [sge_procedures get_standard_job_info], page 188.

See Section 18.18 [sge_procedures get_extended_job_info], page 180.

18.30 get_suspend_state_of_job

NAME

```
get_suspend_state_of_job -- get suspend state of job from ps command
```

SYNOPSIS

```
get_suspend_state_of_job { jobid { pidlist pid_list } {do_error_check 1}
}
```

FUNCTION

```
This procedure returns the suspend state of jobid (letter from ps command)
Beyond that a array (pidlist) is set, in which all process id of the proces
group are listed. The caller of the function can access the array pid_list
```

INPUTS

```
jobid                 - job identification number
{ pidlist pid_list } - name of variable to store the pidlist
{do_error_check 1}   - enable error messages (add_proc_error), default
                       if not 1 the procedure will not report errors
```

RESULT

```
suspend state (letter from ps command)
```

SEE ALSO

See Section 18.20 [sge_procedures get_grppid_of_job], page 182.

See 'sge_procedures/add_proc_error'

18.31 get_version_info

NAME

```
get_version_info -- get version number of the cluster software
```

SYNOPSIS

```
get_version_info { }
```

FUNCTION

```
This procedure will return the version string
```

RESULT

```
returns the first line of "qconf -help" (this is the version number of
the SGEEE/SGE system).
```

SEE ALSO

See '/'

18.32 gethostname

NAME

```
gethostname -- ???
```

SYNOPSIS

```
gethostname { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.33 hold_job

NAME

```
hold_job -- set job in hold state
```

SYNOPSIS

```
hold_job { jobid }
```

FUNCTION

```
This procedure will use the qhold binary to set a job into hold state.
```

INPUTS

```
jobid - job identification number
```

RESULT

```
 0 - ok
-1 - timeout error
```

SEE ALSO

See Section 18.39 [sge_procedures release_job], page 194.

See Section 18.33 [sge_procedures hold_job], page 191.

18.34 is_job_running

NAME

```
is_job_running -- get run information of job
```

SYNOPSIS

```
is_job_running { jobid jobname }
```

FUNCTION

```
This procedure will call qstat -f for job information
```

INPUTS

```
jobid   - job identifaction number
jobname - name of the job (string)
```

RESULT

```
 0 - job is not running (but pending)
 1 - job is running
-1 - not in stat list
```

NOTES

```
This procedure returns 1 (job is running) when the job
is spooled to a queue. This doesn not automatically mean
that the job is "real running".
```

SEE ALSO

See Section 18.34 [sge_procedures is_job_running], page 191.

See Section 18.35 [sge_procedures is_pid_with_name_existing], page 192.

18.35 is_pid_with_name_existing

NAME

```
is_pid_with_name_existing -- search for process on remote host
```

SYNOPSIS

```
is_pid_with_name_existing { host pid proc_name }
```

FUNCTION

```
This procedure will start the checkprog binary with the given parameters.
```

INPUTS

```
host      - remote host
pid       - pid of process
proc_name - process program name
```

RESULT

```
0 - ok; != 0 on error
```

SEE ALSO

See Section 18.34 [sge_procedures is_job_running], page 191.
See Section 18.35 [sge_procedures is_pid_with_name_existing], page 192.

18.36 master_queue_of

NAME

```
master_queue_of -- get the master queue of a parallel job
```

SYNOPSIS

```
master_queue_of { job_id }
```

FUNCTION

```
This procedure will return the name of the master queue of a
parallel job or "" if the MASTER queue was not found.
```

INPUTS

```
job_id - Identification number of the job
```

RESULT

```
empty or the last queue name on which the MASTER task is running
```

SEE ALSO

See Section 18.53 [sge_procedures slave_queue_of], page 206.

18.37 move_qmaster_spool_dir

NAME

```
move_qmaster_spool_dir -- ???
```

SYNOPSIS

```
move_qmaster_spool_dir { new_spool_dir }
```

FUNCTION

```
???
```

INPUTS

```
new_spool_dir - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.38 mqattr

NAME

```
mqattr -- Modify queue attributes
```

SYNOPSIS

```
mqattr { attribute entry queue_list }
```

FUNCTION

```
This procedure enables the caller to modify particular queue attributes.
Look at set_queue for queue attributes.
```

INPUTS

```
attribute  - name of attribute to modify
entry      - new value for attribute
queue_list - name of queues to change
```

RESULT

```
-1 - error
0  - ok
```

EXAMPLE

```
set return_value [mqattr "calendar" "always_disabled" "$queue_list"]
```

SEE ALSO

See Section 18.38 [sge_procedures mqattr], page 193.
See Section 18.47 [sge_procedures set_queue], page 200.
See Section 18.5 [sge_procedures add_queue], page 171.
See Section 18.11 [sge_procedures del_queue], page 175.
See Section 18.26 [sge_procedures get_queue], page 185.
See Section 18.59 [sge_procedures suspend_queue], page 209.
See Section 18.62 [sge_procedures unsuspend_queue], page 211.
See Section 18.13 [sge_procedures disable_queue], page 176.
See Section 18.14 [sge_procedures enable_queue], page 177.

18.39 release_job

NAME

```
release_job -- release job from hold state
```

SYNOPSIS

```
release_job { jobid }
```

FUNCTION

```
This procedure will release the job from hold.
```

INPUTS

```
jobid - job identification number
```

RESULT

```
 0   - ok
-1   - timeout error
```

SEE ALSO

See Section 18.39 [sge_procedures release_job], page 194.
See Section 18.33 [sge_procedures hold_job], page 191.

18.40 reset_schedd_config

NAME

```
reset_schedd_config -- set schedd configuration default values
```

SYNOPSIS

```
reset_schedd_config { }
```

FUNCTION

```
This procedure will call set_schedd_config with default values
```

RESULT

```
-1 : timeout error
 0 : ok
```

NOTES

```
The default values are:

SGE system:

algorithm                    "default"
schedule_interval            "0:0:15"
maxujobs                     "0"
maxgjobs                     "0"
queue_sort_method            "share"
user_sort                    "false"
job_load_adjustments         "np_load_avg=0.50"
load_adjustment_decay_time   "0:7:30"
load_formula                 "np_load_avg"
```

```
schedd_job_info              "true"


SGEEE extensions:

sgeee_schedule_interval        "00:01:00"
halftime                     "0"
usage_weight_list            "cpu=0.34,mem=0.33,io=0.33"
compensation_factor          "5"
weight_user                  "0"
weight_project               "0"
weight_jobclass              "0"
weight_department            "0"
weight_job                   "0"
weight_tickets_functional    "0"
weight_tickets_share         "0"
weight_tickets_deadline      "10000"
```

SEE ALSO

See Section 18.48 [sge_procedures set_schedd_config], page 202.

18.41 resolve_arch

NAME

```
resolve_arch -- ???
```

SYNOPSIS

```
resolve_arch { { host "none" } }
```

FUNCTION

```
???
```

INPUTS

```
{ host "none" } - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.42 resolve_host

NAME

```
resolve_host -- ???
```

SYNOPSIS

```
resolve_host { name { long 0 } }
```

FUNCTION

```
???
```

INPUTS

```
name        - ???
{ long 0 } - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.43 resolve_upper_arch

NAME

```
resolve_upper_arch -- ???
```

SYNOPSIS

```
resolve_upper_arch { host }
```

FUNCTION

```
???
```

INPUTS

```
host - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.44 resolve_version

NAME

```
resolve_version() -- get testsuite internal version number for product
```

SYNOPSIS

```
resolve_version { { internal_number -100 } }
```

FUNCTION

```
This procedure will compare the product version string with known version
numbers of the cluster software. A known version number will return a
value > 0. The return value is an integer and the test procedures can
enable or disable a check procedure by using this number.
If an internal version number is given as parameter, a list of
SGE versions mapping to this internal number is returned.
```

INPUTS

```
{ internal_number -100 } - optional parameter
                           if set to a integer value > -3 the function
                           will return a list of corresponding product
                           version strings.
```

RESULT

```
when internal_number == -100 :
==============================

-4  - unsupported version
-3  - system not running
-2  - system not installed
-1  - unknown error (testsuite error)
 0  - version number not set (testsuite error)
 1  - SGE 5.0.x
 2  - SGEEE 5.0.x
 ...

when internal_number != -100 :
==============================

 List of version strings of the cluster software that match the
 internal version number of the testsuite.
```

KNOWN BUGS

```
A version string should not contain underscores (_); if an internal
version number is given to resolve_version, all underscores are mapped
to a space.
```

SEE ALSO

See Section 18.31 [sge_procedures get_version_info], page 190.

18.45 set_config

NAME

```
set_config -- change global or host specific configuration
```

SYNOPSIS

```
set_config { change_array {host global} }
```

FUNCTION

```
Set the cluster global or exec host local configuration corresponding to
the content of the change_array.
```

INPUTS

```
change_array  - name of an array variable that will be set by get_config
{host global} - set configuration for a specific hostname (host) or set
                the global configuration (global)
```

RESULT

```
-1 : timeout
 0 : ok

The change_array variable is build as follows:

set change_array(xterm)   "/bin/xterm"
set change_array(enforce_project) "true"
...
(every value that is set will be changed)
```

EXAMPLE

```
get_config gcluster1 lobal
set cluster1(qmaster_spool_dir) "/bla/bla/tmp"
set_config cluster1

Here the possible change_array values with some typical settings:

qmaster_spool_dir    /../default/spool/qmaster
execd_spool_dir      /../default/spool
qsi_common_dir       /../default/common/qsi
binary_path          /../bin
mailer               /usr/sbin/Mail
xterm                /usr/bin/X11/xterm
load_sensor          none
prolog               none
epilog               none
shell_start_mode     posix_compliant
login_shells         sh,ksh,csh,tcsh
min_uid              0
min_gid              0
user_lists           none
xuser_lists          none
projects             none
xprojects            none
load_report_time     00:01:00
stat_log_time        12:00:00
max_unheard          00:02:30
loglevel             log_info
```

```
enforce_project      false
administrator_mail   none
set_token_cmd        none
pag_cmd              none
token_extend_time    none
shepherd_cmd         none
qmaster_params       none
schedd_params        none
execd_params         none
finished_jobs        0
gid_range            13001-13100
admin_user           crei
qlogin_command       telnet
qlogin_daemon        /usr/etc/telnetd
```

SEE ALSO

See Section 18.15 [sge_procedures get_config], page 177.

18.46 set_execHost

NAME

```
set_exechost -- set/change exec host configuration
```

SYNOPSIS

```
set_exechost { change_array host }
```

FUNCTION

```
Set the exec host configuration corresponding to the content of the
change_array.
```

INPUTS

```
change_array - name of an array variable that will be set by set_exechost
host         - name of an execution host
```

RESULT

```
The array should look like follows:

set change_array(user_list)   "deadlineusers"
set change_array(load_scaling) "NONE"
....
(every value that is set will be changed)


Here the possible change_array values with some typical settings:

hostname                    myhost.mydomain
load_scaling                NONE
complex_list                test
complex_values              NONE
user_lists                  deadlineusers
xuser_lists                 NONE
projects                    NONE
xprojects                   NONE
```

```
usage_scaling              NONE
resource_capability_factor 0.000000

return value:
-100 :    unknown error
-1   :    on timeout
   0 :    ok
```

EXAMPLE

```
get_exechost myconfig expo1
set myconfig(user_lists) NONE
set_exechost myconfig expo1
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 18.17 [sge_procedures get_exechost], page 179.

18.47 set_queue

NAME

```
set_queue -- set or change queue configuration
```

SYNOPSIS

```
set_queue { q_name change_array }
```

FUNCTION

```
Set a queue configuration corresponding to the content of the change_array
```

INPUTS

```
q_name       - name of the queue to configure
change_array - name of an array variable that will be set by set_queue
```

RESULT

```
0  : ok
-1 : timeout
```

EXAMPLE

```
get_queue myqueue.q queue1
set queue1(load_thresholds) "np_load_avg=3.75"
set_queue myqueue.q queue1
```

NOTES

```
the array should look like this:

set change_array(qname) MYHOST
set change_array(hostname) MYHOST.domain
....
(every value that is set will be changed)
```

```
here is a list of all guilty array names (template queue):

change_array(qname)                 "template"
change_array(hostname)              "unknown"
change_array(seq_no)                "0"
change_array(load_thresholds)       "np_load_avg=1.75"
change_array(suspend_thresholds)    "NONE"
change_array(nsuspend)              "0"
change_array(suspend_interval)      "00:05:00"
change_array(priority)              "0"
change_array(max_migr_time)         "0"
change_array(migr_load_thresholds) "np_load_avg=5.00"
change_array(max_no_migr)           "00:02:00"
change_array(min_cpu_interval)      "00:05:00"
change_array(processors)            "UNDEFINED"
change_array(qtype)                 "BATCH INTERACTIVE"
change_array(rerun)                 "FALSE"
change_array(slots)                 "1"
change_array(tmpdir)                "/tmp"
change_array(shell)                 "/bin/csh"
change_array(shell_start_mode)      "NONE"
change_array(klog)                  "/usr/local/bin/klog"
change_array(prolog)                "NONE"
change_array(epilog)                "NONE"
change_array(starter_method)        "NONE"
change_array(suspend_method)        "NONE"
change_array(resume_method)         "NONE"
change_array(terminate_method)      "NONE"
change_array(reauth_time)           "01:40:00"
change_array(notify)                "00:00:60"
change_array(owner_list)            "NONE"
change_array(user_lists)            "NONE"
change_array(xuser_lists)           "NONE"
change_array(subordinate_list)      "NONE"
change_array(complex_list)          "NONE"
change_array(complex_values)        "NONE"
change_array(projects)              "NONE"
change_array(xprojects)             "NONE"
change_array(calendar)              "NONE"
change_array(initial_state)         "default"
change_array(fshare)                "0"
change_array(oticket)               "0"
change_array(s_rt)                  "INFINITY"
change_array(h_rt)                  "INFINITY"
change_array(s_cpu)                 "INFINITY"
change_array(h_cpu)                 "INFINITY"
change_array(s_fsize)               "INFINITY"
change_array(h_fsize)               "INFINITY"
change_array(s_data)                "INFINITY"
change_array(h_data)                "INFINITY"
change_array(s_stack)               "INFINITY"
change_array(h_stack)              "INFINITY"
change_array(s_core)               "INFINITY"
change_array(h_core)               "INFINITY"
change_array(s_rss)                "INFINITY"
change_array(h_rss)                "INFINITY"
change_array(s_vmem)               "INFINITY"
change_array(h_vmem)               "INFINITY"
```

SEE ALSO

See Section 18.38 [sge_procedures mqattr], page 193.
See Section 18.47 [sge_procedures set_queue], page 200.
See Section 18.5 [sge_procedures add_queue], page 171.
See Section 18.11 [sge_procedures del_queue], page 175.
See Section 18.26 [sge_procedures get_queue], page 185.
See Section 18.59 [sge_procedures suspend_queue], page 209.
See Section 18.62 [sge_procedures unsuspend_queue], page 211.
See Section 18.13 [sge_procedures disable_queue], page 176.
See Section 18.14 [sge_procedures enable_queue], page 177.

18.48 set_schedd_config

NAME

```
set_schedd_config -- change scheduler configuration
```

SYNOPSIS

```
set_schedd_config { change_array }
```

FUNCTION

```
Set the scheduler configuration corresponding to the content of the
change_array.
```

INPUTS

```
change_array - name of an array variable that will be set by
               set_schedd_config
```

RESULT

```
-1 : timeout
 0 : ok
```

EXAMPLE

```
get_schedd_config myconfig
set myconfig(schedule_interval) "0:0:10"
set_schedd_config myconfig
```

NOTES

```
The array should be build like follows:

set change_array(algorithm) default
set change_array(schedule_interval) 0:0:15
....
(every value that is set will be changed)
```

```
Here the possible change_array values with some typical settings:

algorithm                    "default"
schedule_interval            "0:0:15"
maxujobs                     "0"
maxgjobs                     "0"
queue_sort_method            "share"
user_sort                    "false"
job_load_adjustments         "np_load_avg=0.50"
load_adjustment_decay_time   "0:7:30"
load_formula                 "np_load_avg"
schedd_job_info              "true"


In case of a SGEEE - System:

sgeee_schedule_interval        "00:01:00"
halftime                     "0"
usage_weight_list            "cpu=0.34,mem=0.33,io=0.33"
compensation_factor          "5"
weight_user                  "0"
weight_project               "0"
weight_jobclass              "0"
weight_department            "0"
weight_job                   "0"
weight_tickets_functional    "0"
weight_tickets_share         "0"
weight_tickets_deadline      "10000"
```

SEE ALSO

See Section 18.28 [sge_procedures get_schedd_config], page 187.

18.49 shutdown_all_shadowd

NAME

```
shutdown_all_shadowd -- ???
```

SYNOPSIS

```
shutdown_all_shadowd { hostname }
```

FUNCTION

```
???
```

INPUTS

```
hostname - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 18.50 [sge_procedures shutdown_core_system], page 204.
See Section 18.51 [sge_procedures shutdown_master_and_scheduler], page 205.
See Section 18.49 [sge_procedures shutdown_all_shadowd], page 203.
See Section 18.52 [sge_procedures shutdown_system_daemon], page 205.
See Section 18.55 [sge_procedures startup_qmaster], page 207.
See Section 18.54 [sge_procedures startup_execd], page 206.
See Section 18.56 [sge_procedures startup_shadowd], page 208.

18.50 shutdown_core_system

NAME

```
shutdown_core_system -- ???
```

SYNOPSIS

```
shutdown_core_system { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 18.50 [sge_procedures shutdown_core_system], page 204.
See Section 18.51 [sge_procedures shutdown_master_and_scheduler], page 205.
See Section 18.49 [sge_procedures shutdown_all_shadowd], page 203.
See Section 18.52 [sge_procedures shutdown_system_daemon], page 205.
See Section 18.55 [sge_procedures startup_qmaster], page 207.
See Section 18.54 [sge_procedures startup_execd], page 206.
See Section 18.56 [sge_procedures startup_shadowd], page 208.

18.51 shutdown_master_and_scheduler

NAME

```
shutdown_master_and_scheduler -- ???
```

SYNOPSIS

```
shutdown_master_and_scheduler { hostname qmaster_spool_dir }
```

FUNCTION

```
???
```

INPUTS

```
hostname          - ???
qmaster_spool_dir - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 18.50 [sge_procedures shutdown_core_system], page 204.
See Section 18.51 [sge_procedures shutdown_master_and_scheduler], page 205.
See Section 18.49 [sge_procedures shutdown_all_shadowd], page 203.
See Section 18.52 [sge_procedures shutdown_system_daemon], page 205.
See Section 18.55 [sge_procedures startup_qmaster], page 207.
See Section 18.54 [sge_procedures startup_execd], page 206.
See Section 18.56 [sge_procedures startup_shadowd], page 208.

18.52 shutdown_system_daemon

NAME

```
shutdown_system_daemon -- kill running sge daemon
```

SYNOPSIS

```
shutdown_system_daemon { host type }
```

FUNCTION

```
This procedure will kill all commd, execd, qmaster or sched processes on
the given host. It does not matter weather the system is sgeee or sge
(sge or sgeee).
```

INPUTS

```
host     - remote host
typelist - list of processes to kill (commd, execd, qmaster or sched)
```

RESULT

```
none
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 18.50 [sge_procedures shutdown_core_system], page 204.
See Section 18.51 [sge_procedures shutdown_master_and_scheduler], page 205.
See Section 18.49 [sge_procedures shutdown_all_shadowd], page 203.
See Section 18.52 [sge_procedures shutdown_system_daemon], page 205.
See Section 18.55 [sge_procedures startup_qmaster], page 207.
See Section 18.54 [sge_procedures startup_execd], page 206.
See Section 18.56 [sge_procedures startup_shadowd], page 208.

18.53 slave_queue_of

NAME

```
slave_queue_of -- Get the last slave queue of a parallel job
```

SYNOPSIS

```
slave_queue_of { job_id }
```

FUNCTION

```
This procedure will return the name of the last slave queue of a
parallel job or "" if the SLAVE queue was not found.
```

INPUTS

```
job_id - Identification number of the job
```

RESULT

```
empty or the last queue name on which the SLAVE task is running
```

SEE ALSO

See Section 18.36 [sge_procedures master_queue_of], page 192.

18.54 startup_execd

NAME

```
startup_execd -- ???
```

SYNOPSIS

```
startup_execd { hostname }
```

FUNCTION

```
???
```

INPUTS

```
hostname - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 18.50 [sge_procedures shutdown_core_system], page 204.
See Section 18.51 [sge_procedures shutdown_master_and_scheduler], page 205.
See Section 18.49 [sge_procedures shutdown_all_shadowd], page 203.
See Section 18.52 [sge_procedures shutdown_system_daemon], page 205.
See Section 18.55 [sge_procedures startup_qmaster], page 207.
See Section 18.54 [sge_procedures startup_execd], page 206.
See Section 18.56 [sge_procedures startup_shadowd], page 208.

18.55 startup_qmaster

NAME

```
startup_qmaster -- ???
```

SYNOPSIS

```
startup_qmaster { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 18.50 [sge_procedures shutdown_core_system], page 204.
See Section 18.51 [sge_procedures shutdown_master_and_scheduler], page 205.
See Section 18.49 [sge_procedures shutdown_all_shadowd], page 203.
See Section 18.52 [sge_procedures shutdown_system_daemon], page 205.
See Section 18.55 [sge_procedures startup_qmaster], page 207.
See Section 18.54 [sge_procedures startup_execd], page 206.
See Section 18.56 [sge_procedures startup_shadowd], page 208.

18.56 startup_shadowd

NAME

```
startup_shadowd -- ???
```

SYNOPSIS

```
startup_shadowd { hostname }
```

FUNCTION

```
???
```

INPUTS

```
hostname - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 18.50 [sge_procedures shutdown_core_system], page 204.
See Section 18.51 [sge_procedures shutdown_master_and_scheduler], page 205.
See Section 18.49 [sge_procedures shutdown_all_shadowd], page 203.
See Section 18.52 [sge_procedures shutdown_system_daemon], page 205.
See Section 18.55 [sge_procedures startup_qmaster], page 207.
See Section 18.54 [sge_procedures startup_execd], page 206.
See Section 18.56 [sge_procedures startup_shadowd], page 208.

18.57 submit_job

NAME

```
submit_job -- submit a job with qsub
```

SYNOPSIS

```
submit_job { args {do_error_check 1} {submit_timeout 30} }
```

FUNCTION

```
This procedure will submit a job.
```

INPUTS

```
args                 - a string of qsub arguments/parameters
{do_error_check 1}   - if 1 (default): add global erros (add_proc_error)
                       if not 1: do not add errors
{submit_timeout 30} - timeout (default is 30 sec.)
```

RESULT

```
This procedure returns:

jobid   of array or job if submit was successfull (value > 1)
   -1   on timeout error
   -2   if usage was printed on -help or commandfile argument
   -3   if usage was printed NOT on -help or commandfile argument
   -4   if verify output was printed on -verify argument
   -5   if verify output was NOT printed on -verfiy argument
   -6   job could not be scheduled, try later
 -100   on error
```

EXAMPLE

```
set jobs ""
set my_outputs "-o /dev/null -e /dev/null"
set arguments "$my_outputs -q $rerun_queue -r y $CHECK_PRODUCT_ROOT/exampl
lappend jobs [submit_job $arguments]
```

SEE ALSO

See Section 18.12 [sge_procedures delete_job], page 176.
See Section 1.1 [check add_proc_error], page 1.

18.58 suspend_job

NAME

```
suspend_job -- set job in suspend state
```

SYNOPSIS

```
suspend_job { id }
```

FUNCTION

```
This procedure will call qmod to suspend the given job id.
```

INPUTS

```
id - job identification number
```

RESULT

```
0  - ok
-1 - error
```

SEE ALSO

See Section 18.61 [sge_procedures unsuspend_job], page 211.

18.59 suspend_queue

NAME

```
suspend_queue -- set a queue in suspend mode
```

SYNOPSIS

```
suspend_queue { qname }
```

FUNCTION

```
This procedure will set the given queue into suspend state
```

INPUTS

```
qname - name of the queue to suspend
```

RESULT

```
 0  - ok
-1  - error
```

SEE ALSO

See Section 18.38 [sge_procedures mqattr], page 193.
See Section 18.47 [sge_procedures set_queue], page 200.
See Section 18.5 [sge_procedures add_queue], page 171.
See Section 18.11 [sge_procedures del_queue], page 175.
See Section 18.26 [sge_procedures get_queue], page 185.
See Section 18.59 [sge_procedures suspend_queue], page 209.
See Section 18.62 [sge_procedures unsuspend_queue], page 211.
See Section 18.13 [sge_procedures disable_queue], page 176.
See Section 18.14 [sge_procedures enable_queue], page 177.

18.60 test

NAME

```
test -- ???
```

SYNOPSIS

```
test { m p }
```

FUNCTION

```
???
```

INPUTS

```
m - ???
p - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

18.61 unsuspend_job

NAME

```
unsuspend_job -- set job bakr from unsuspended state
```

SYNOPSIS

```
unsuspend_job { job }
```

FUNCTION

```
This procedure will call qmod to unsuspend the given job id.
```

INPUTS

```
job - job identification number
```

RESULT

```
0   - ok
-1  - error
```

SEE ALSO

See Section 18.58 [sge_procedures suspend_job], page 209.

18.62 unsuspend_queue

NAME

```
unsuspend_queue -- set a queue in suspend mode
```

SYNOPSIS

```
unsuspend_queue { queue }
```

FUNCTION

```
This procedure will set the given queue into unsuspend state
```

INPUTS

```
queue - name of the queue to set into unsuspend state
```

RESULT

```
 0  - ok
-1  - error
```

SEE ALSO

See Section 18.38 [sge_procedures mqattr], page 193.
See Section 18.47 [sge_procedures set_queue], page 200.
See Section 18.5 [sge_procedures add_queue], page 171.
See Section 18.11 [sge_procedures del_queue], page 175.
See Section 18.26 [sge_procedures get_queue], page 185.
See Section 18.59 [sge_procedures suspend_queue], page 209.
See Section 18.62 [sge_procedures unsuspend_queue], page 211.
See Section 18.13 [sge_procedures disable_queue], page 176.
See Section 18.14 [sge_procedures enable_queue], page 177.

18.63 wait_for_end_of_all_jobs

NAME

```
wait_for_end_of_all_jobs() -- wait for end of all jobs
```

SYNOPSIS

```
wait_for_end_of_all_jobs { seconds }
```

FUNCTION

```
This procedure will wait until no further jobs are remaining in the cluste
```

INPUTS

```
seconds - timeout value (if < 1 no timeout is set)
```

RESULT

```
 0 - ok
-1 - timeout
```

SEE ALSO

See Section 18.65 [sge_procedures wait_for_jobend], page 213.

18.64 wait_for_end_of_transfer

NAME

```
wait_for_end_of_transfer -- wait transfer end of job
```

SYNOPSIS

```
wait_for_end_of_transfer { jobid seconds }
```

FUNCTION

```
This procedure will parse the qstat output of the job for the t state. If
no t state is found for the given job id, the procedure will return.
```

INPUTS

```
jobid   - job identification number
seconds - timeout in seconds
```

RESULT

```
 0 - job is not in transferstate
-1 - timeout
```

EXAMPLE

```
see "sge_procedures/wait_for_jobstart"
```

SEE ALSO

See Section 18.68 [sge_procedures wait_for_load_from_all_queues], page 215.
See Section 3.12 [file_procedures wait_for_file], page 46.
See Section 18.67 [sge_procedures wait_for_jobstart], page 214.
See Section 18.64 [sge_procedures wait_for_end_of_transfer], page 212.
See Section 18.66 [sge_procedures wait_for_jobpending], page 213.
See Section 18.65 [sge_procedures wait_for_jobend], page 213.

18.65 wait_for_jobend

NAME

```
wait_for_jobend -- wait for end of job
```

SYNOPSIS

```
wait_for_jobend { jobid jobname seconds }
```

FUNCTION

```
This procedure is testing first if the given job is really running. After
that it waits for the job to disappear in the qstat output.
```

INPUTS

```
jobid   - job identification number
jobname - name of job
seconds - timeout in seconds
```

RESULT

```
 0 - job stops running
-1 - timeout error
-2 - job is not running
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See Section 18.63 [sge_procedures wait_for_end_of_all_jobs], page 212.
See Section 18.68 [sge_procedures wait_for_load_from_all_queues], page 215.
See Section 3.12 [file_procedures wait_for_file], page 46.
See Section 18.67 [sge_procedures wait_for_jobstart], page 214.
See Section 18.64 [sge_procedures wait_for_end_of_transfer], page 212.
See Section 18.66 [sge_procedures wait_for_jobpending], page 213.
See Section 18.65 [sge_procedures wait_for_jobend], page 213.

18.66 wait_for_jobpending

NAME

```
wait_for_jobpending -- wait for job to get into pending state
```

SYNOPSIS

```
wait_for_jobpending { jobid jobname seconds }
```

FUNCTION

```
This procedure will return when the job is in pending state.
```

INPUTS

```
jobid   - job identification number
jobname - name of the job
seconds - timeout value in seconds
```

RESULT

```
-1  on timeout
0   when job is in pending state
```

EXAMPLE

```
foreach elem $sched_jobs {
    wait_for_jobpending $elem "Sleeper" 300
}
```

SEE ALSO

See Section 18.68 [sge_procedures wait_for_load_from_all_queues], page 215.
See Section 3.12 [file_procedures wait_for_file], page 46.
See Section 18.67 [sge_procedures wait_for_jobstart], page 214.
See Section 18.64 [sge_procedures wait_for_end_of_transfer], page 212.
See Section 18.66 [sge_procedures wait_for_jobpending], page 213.
See Section 18.65 [sge_procedures wait_for_jobend], page 213.

18.67 wait_for_jobstart

NAME

```
wait_for_jobstart -- wait for job to get out of pending list
```

SYNOPSIS

```
wait_for_jobstart { jobid jobname seconds {do_errorcheck 1} }
```

FUNCTION

```
This procedure will call the is_job_running procedure in a while
loop. When the job is scheduled to a queue the job is "running"
and the procedure returns.
```

INPUTS

```
jobid             - job identification number
jobname           - name of the job
seconds           - timeout in seconds
{do_errorcheck 1} - enable error check (default)
                    if 0: do not report errors
```

RESULT

```
-1 - job is not running (timeout error)
 0 - job is running ( not in pending state)
```

EXAMPLE

```
foreach elem $jobs {
   wait_for_jobstart $elem "Sleeper" 300
   wait_for_end_of_transfer $elem 300
   append jobs_string "$elem "
}
```

SEE ALSO

See Section 18.68 [sge_procedures wait_for_load_from_all_queues], page 215.
See Section 3.12 [file_procedures wait_for_file], page 46.
See Section 18.67 [sge_procedures wait_for_jobstart], page 214.
See Section 18.64 [sge_procedures wait_for_end_of_transfer], page 212.
See Section 18.66 [sge_procedures wait_for_jobpending], page 213.
See Section 18.65 [sge_procedures wait_for_jobend], page 213.

18.68 wait_for_load_from_all_queues

NAME

```
wait_for_load_from_all_queues -- wait for load value reports from queues
```

SYNOPSIS

```
wait_for_load_from_all_queues { seconds }
```

FUNCTION

```
This procedure waits until all queues are reporting a load value smaller
than 99. If this is the case all execd should be successfully connected
to the qmaster.
```

INPUTS

```
seconds - timeout value in seconds
```

RESULT

```
"-1" on error
```

SEE ALSO

See Section 18.68 [sge_procedures wait_for_load_from_all_queues], page 215.
See Section 3.12 [file_procedures wait_for_file], page 46.
See Section 18.67 [sge_procedures wait_for_jobstart], page 214.
See Section 18.64 [sge_procedures wait_for_end_of_transfer], page 212.
See Section 18.66 [sge_procedures wait_for_jobpending], page 213.
See Section 18.65 [sge_procedures wait_for_jobend], page 213.

18.69 was_job_running

NAME

```
was_job_running -- look for job accounting
```

SYNOPSIS

```
was_job_running { jobid {do_errorcheck 1} }
```

FUNCTION

```
This procedure will start a qacct -j jobid. If the hob was not found in
the output of the qacct command, this function will return -1. This
means that the job is still running, or was never running.
```

INPUTS

```
jobid             - job identification number
{do_errorcheck 1} - 1: call add_proc_error if job was not found
                    0: do not generate error messages
```

RESULT

```
"-1"  : if job was not found
or the output of qacct -j
```

SEE ALSO

See Section 1.1 [check add_proc_error], page 1.

19 size

19.1 check_flood

NAME

```
check_flood -- ???
```

SYNOPSIS

```
check_flood { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

19.2 check_idle

NAME

```
check_idle -- ???
```

SYNOPSIS

```
check_idle { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

19.3 check_miniworm

NAME

```
check_miniworm -- ???
```

SYNOPSIS

```
check_miniworm { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

19.4 check_qstat

NAME

```
check_qstat -- ???
```

SYNOPSIS

```
check_qstat { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

19.5 check_size_cleanup

NAME

```
check_size_cleanup -- ???
```

SYNOPSIS

```
check_size_cleanup { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

19.6 check_size_config

NAME

```
check_size_config -- ???
```

SYNOPSIS

```
check_size_config { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

19.7 check_size_config_zombies

NAME

```
check_size_config_zombies -- ???
```

SYNOPSIS

```
check_size_config_zombies { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

19.8 get_job_count

NAME

```
get_job_count -- ???
```

SYNOPSIS

```
get_job_count { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

19.9 get_last_jobid

NAME

```
get_last_jobid -- ???
```

SYNOPSIS

```
get_last_jobid { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

19.10 get_size

NAME

```
get_size -- ???
```

SYNOPSIS

```
get_size { who }
```

FUNCTION

```
???
```

INPUTS

```
who - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

19.11 init_level

NAME

`init_level -- ???`

SYNOPSIS

`init_level { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See '/'

19.12 init_ps

NAME

`init_ps -- ???`

SYNOPSIS

`init_ps { }`

FUNCTION

`???`

RESULT

`???`

EXAMPLE

`???`

NOTES

`???`

BUGS

`???`

SEE ALSO

See '/'

19.13 monitor

NAME

```
monitor -- ???
```

SYNOPSIS

```
monitor { text duration interval commands }
```

FUNCTION

```
???
```

INPUTS

```
text     - ???
duration - ???
interval - ???
commands - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

19.14 monitor_header

NAME

```
monitor_header -- ???
```

SYNOPSIS

```
monitor_header { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

19.15 monitor_size

NAME

```
monitor_size -- ???
```

SYNOPSIS

```
monitor_size { jobs {when ""} }
```

FUNCTION

```
???
```

INPUTS

```
jobs      - ???
{when ""} - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

19.16 output_monitor_result

NAME

```
output_monitor_result -- ???
```

SYNOPSIS

```
output_monitor_result { start_size end_size }
```

FUNCTION

```
???
```

INPUTS

```
start_size - ???
end_size   - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

19.17 qstat_test

NAME

```
qstat_test -- ???
```

SYNOPSIS

```
qstat_test { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

19.18 restart_system

NAME

```
restart_system -- ???
```

SYNOPSIS

```
restart_system { }
```

FUNCTION

```
???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See '/'

19.19 stabilize

NAME

```
stabilize -- ???
```

SYNOPSIS

```
stabilize { text delay interval commands }
```

FUNCTION

```
???
```

INPUTS

```
text     - ???
delay    - ???
interval - ???
commands - ???
```

RESULT

```
???
```

EXAMPLE

```
???
```

NOTES

```
???
```

BUGS

```
???
```

SEE ALSO

See ‘/’

Function Index

Table of Contents