

# Function Reference

Simple Grid Protocol version: 1.2

## Contents:

- p.2 - Function: porta
- p.3 - Function: sgp-make-thread
- p.4 - Function: sgp-join-threads

# Function: porta

**Simple Grid Protocol version:** 1.2

**File:** porta.lisp

**Declared argument list:** (task &optional (nth\_best\_ip 0) (return\_list NIL))

**Usage:** (porta task nth\_best\_ip return\_list)

## Arguments:

- task - *Required, Quoted List*, a single lisp form or object to be evaluated and the result returned.
- nth\_best\_ip - *Optional, Integer, Default=0*, N sends task to computer with Nth best ACR (0 for best ACR, 1 for next best ACR and so forth) for evaluation.
- return\_list - *Optional, T or NIL, Default=NIL*, T returns a list instead of values.

## Description:

Porta function makes a programming task portable by obtaining a list of computer IP addresses on the grid from the Tracking Server, and then the sends the task to the grid for execution on the machine with the highest ACR (default) or Nth best ACR (N is 0 for best ACR, 1 for next best ACR and so forth). Task will be self executed if local machine has the same IP as the grid machine executing the task.

$ACR = CPU\_Frequency(MHz) \times CPU\_Idle(\%) / 100$

NOTE: Porta uses an IP rotary system. If there are less computers on the grid than required by porta, porta will loop back to the computer (IP address) with the best ACR after the computer (IP address) with the worst ACR.

IP rotary system example for tasks sent to 4 best computers when the grid has only 3 computers:

```
(porta '(* 3 4) 0) ==to best IP==>
(porta '(* 2 5) 1) ==to next best IP==>
(porta '(* 3 2) 2) ==to worst IP==>
(porta '(* 5 4) 3) ==to best IP==>
```

## Returned values:

- \* Result of portable task execution
  - \* IP Address of machine that executed portable task
  - \* T (if self executed) or NIL (if grid executed)
- or an optional list containing the above values.

# Function: sgp-make-thread

**Simple Grid Protocol version:** 1.2

**File:** sgp\_thread.lisp

**Declared argument list:** (anylist &optional (main\_thread\_id 0))

**Usage:** (sgp-make-thread anylist main\_thread\_id)

## Arguments:

anylist - *Required, Quoted List*, a single lisp form or object to be evaluated in a new thread.

main\_thread\_id - *Optional, Integer, Default=0*, Thread ID of the main program making this thread.

## Description:

Evaluates anylist in a new Operating System thread.

## Returned value:

\* *Integer*, Thread ID of the new thread.

# Function: sgp-join-threads

**Simple Grid Protocol version:** 1.2

**File:** sgp\_thread.lisp

**Declared argument list:** (threads &optional (main\_thread\_id 0))

**Usage:** (sgp-join-threads threads main\_thread\_id)

## Arguments:

**threads** - *Required, List*, Thread IDs of threads made by function sgp-make-thread.  
**main\_thread\_id** - *Optional, Integer, Default=0*, Thread ID of the main program that made the threads being joined.

## Description:

When all Operating System threads (specified in the threads list argument) have completed evaluating their corresponding list, this function returns a list of each evaluation and it's thread\_id. The main\_thread\_id argument value must be identical to that used by function sgp-make-thread for the threads to be joined.

## Returned value:

\* *List*, containing multiple lists of evaluation and thread\_id pairs. Format (for 3 threads):  
( (evaluation thread\_id) (evaluation thread\_id) (evaluation thread\_id) )