

Class ThreadGroup < Object

A ThreadGroup keeps track of a number of threads. A Thread can belong to only one ThreadGroup at a time; adding a thread to a group will remove it from its current group. Newly created threads belong to the group of the thread that created them.

ThreadGroup constants

Default Default thread group.

Class methods

new ThreadGroup.new → *thgrp*
Returns a newly created ThreadGroup. The group is initially empty.

Instance methods

add *thgrp.add(thread)* → *thgrp*
Adds the given thread to this group, removing it from any other group to which it may have previously belonged.

```
puts "Default group is #{ThreadGroup::Default.list}"
tg = ThreadGroup.new
t1 = Thread.new { sleep }
t2 = Thread.new { sleep }
puts "t1 is #{t1}, t2 is #{t2}"
tg.add(t1)
puts "Default group now #{ThreadGroup::Default.list}"
puts "tg group now #{tg.list}"
```

produces:

```
Default group is [#<Thread:0x0ac684 run>]
t1 is #<Thread:0x0a35fc>, t2 is #<Thread:0x0a34f8>
Default group now [#<Thread:0x0ac684 run>, #<Thread:0x0a34f8 run>]
tg group now [#<Thread:0x0a35fc run>]
```

enclose *thgrp.enclose* → *thgrp*

1.9 Prevents threads being added to and removed from *thgrp*. New threads may still be started.

```
thread = Thread.new { sleep 99 }
group = ThreadGroup.new
group.add(thread)
group.enclose
ThreadGroup::Default.add(thread)
```

produces:

```
prog.rb:5:in `add': can't move from the enclosed thread group (ThreadError)
from /tmp/prog.rb:5:in `<main>'
```

enclosed? *thgrp.enclose* → true or false

1.9 Returns true if this thread group has been enclosed.

list*thgrp.list* → *array*

Returns an array of all existing Thread objects that belong to this group.

```
ThreadGroup::Default.list # => [#<Thread:0x0ac684 run>]
```