

Limits

[Previous](#) [Up](#) [Next](#)

Limits provide simple load management by limiting the number of tasks submitted by a specific [ecflow_server](#)

We have learnt from experience that suite designers were using [trigger](#)'s in two different ways: as data dependency triggers and as courtesy triggers.

Triggers were designed for the former. The latter are used to prevent too many jobs running at once and are actually an artificial way of queuing jobs.

Because ecFlow does not distinguish between the two sorts of triggers, suites can become difficult to maintain after a while. So the concept of [limit](#) was introduced.

Limits are declared with the **limit** keyword

inlimit

Limits are used in conjunction with [inlimit](#) keyword.

First, a [limit](#) must be defined using the 'limit NAME N'.

The limit definition is typically placed at the [suite](#) scope.

Next we create a group of tasks to which we want to apply the limit.

This is done by attaching an 'inlimit NAME' attribute to the nodes.

Attaching the attribute to a [task](#) adds the task to the group.

Attaching it to a [family](#) adds all tasks from that [family](#).

The effect of a [limit](#) is that no more than N tasks of a group will run at once.

A [node](#) can be limited by several limits.

Ecf script

We will create [family](#) f5 with nine tasks.

Create new [ecf script](#) s in \$HOME/course/test/f5/ directory, each one containing:

```
$HOME/course/test/f5/t1.ecf,t2.ecf.....t9.ecf
```

```
%include <head.h>
echo "I will now sleep for %SLEEP% seconds"
sleep %SLEEP%
%include <tail.h>
```

Text

Let us modify our [suite definition](#) file:

```
# Definition of the suite test.
suite test
  edit ECF_INCLUDE "$HOME/course"
  edit ECF_HOME    "$HOME/course"
  limit l1 2

  family f5
    inlimit l1
    edit SLEEP 20
    task t1
    task t2
    task t3
    task t4
    task t5
    task t6
    task t7
    task t8
    task t9
  endfamily
endsuite
```

Python

\$HOME/course/test.py

```
import os
from ecflow import Defs, Suite, Family, Task, Edit, Trigger, Complete, Event, Meter, Time, Day, Date, Label, \
    RepeatString, RepeatInteger, RepeatDate, InLimit, Limit

def create_family_f5() :
    return Family("f5",
        InLimit("l1"),
        Edit(SLEEP=20),
        [ Task('t{}'.format(i)) for i in range(1,10) ] )

print("Creating suite definition")
home = os.path.join(os.getenv("HOME"), "course")
defs = Defs(
    Suite("test",
        Edit(ECF_INCLUDE=home, ECF_HOME=home),
        Limit("l1", 2),
        create_family_f5()))
print(defs)

print("Checking job creation: .ecf -> .job0")
print(defs.check_job_creation())

print("Checking trigger expressions and inlimits")
assert len(defs.check()) == 0, defs.check()

print("Saving definition to file 'test.def'")
defs.save_as_defs("test.def")
```

What to do

1. Edit the changes
2. Replace the [suite definition](#)
3. In [ecflow_ui](#), observe the triggers of the [limit l1](#)
4. Open the Info panel for **l1**
5. Change the value of the [limit](#)
6. Open the Why? panel for one of the [queued](#) tasks of **/test/f5**
7. Introduce an error in the limits and make sure this error is trapped. i.e. change the Limit.

Check InLimit/Limit references

```
Limit("unknown", 2)
```

[Previous](#) [Up](#) [Next](#)