

Indentation

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

The text examples allow indentation. In python, indentation affects the logical meaning of the program.

However, as we have seen we can use the Node constructor to provide indentation, to show the definition structure.

Using Node constructor to show definition structure

```
import os
from ecflow import Defs,Suite,Family,Task,Edit,Trigger,Complete,Event,Meter,Time,Day,Date

print("Creating suite definition")
home = os.path.join(os.getenv("HOME"), "course")
defs = Defs(
    Suite("test",
        Edit(ECF_INCLUDE=home,ECF_HOME=home),
        Family("f1",
            Edit(SLEEP=20),
            Task("t1", Meter("progress", 1, 100, 90)),
            Task("t2", Trigger("t1 == complete"),Event("a"),Event("b")),
            Task("t3", Trigger("t2:a")),
            Task("t4", Trigger("t2 == complete"), Complete("t2:b")),
            Task("t5", Trigger("t1:progress ge 30")),
            Task("t6", Trigger("t1:progress ge 60")),
            Task("t7", Trigger("t1:progress ge 90")),
        Family("f2",
            Edit(SLEEP=20),
            Task("t1", Time( "00:30 23:30 00:30" )),
            Task("t2", Day( "sunday" )),
            Task("t3", Date("1.*.*"), Time("12:00")),
            Task("t4", Time("+00:02")),
            Task("t5", Time("00:02")))))
print(defs)

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

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

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

We can also use python **with** statement to provide the indentation.

Here is the previous example using the **with** statement:

Using With

```
from ecflow import Defs,Suite,Family,Task,Edit,Trigger,Complete,Event,Meter,Time,Day,Date
import os
import sys

print("Creating suite definition")
with Defs() as defs:

    with defs.add_suite("test") as suite:
        suite += Edit(ECF_HOME=os.path.join(os.getenv("HOME"), "course"))
        suite += Edit(ECF_INCLUDE =os.path.join(os.getenv("HOME"), "course"))

    with suite.add_family("f1") as f1:
        f1 += Edit(SLEEP=20)
        f1 += Task("t1", Meter("progress", 1, 100, 90))
        f1 += Task("t2", Trigger("t1 == complete"), Event("a"), Event("b"))
        f1 += Task("t3", Trigger("t2:a"))
        f1 += Task("t4", Trigger("t2 == complete"), Complete("t2:b"))
        f1 += Task("t5", Trigger("t1:progress ge 30"))
        f1 += Task("t6", Trigger("t1:progress ge 60"))
        f1 += Task("t7", Trigger("t1:progress ge 90"))

    with suite.add_family("f2") as f2:
        f2 += Edit(SLEEP=20)
        f2 += Task("t1", Time("00:30 23:30 00:30"))
        f2 += Task("t2", Day("sunday"))
        f2 += Task("t3", Date(1, 0, 0), Time(12, 0))
        f2 += Task("t4", Time(0, 2, True))
        f2 += Task("t5", Time(0, 2))

print(defs)

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

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

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

[Previous Up Next](#)

Alternative styles:

```

import os
for ecflow import *

print("Creating suite definition")
home = os.path.join(os.getenv("HOME"), "course")
with Defs() as defs:
    with defs.add_suite("test") as suite:
        suite += Edit(ECF_INCLUDE=home, ECF_HOME=home)
        with suite.add_family("f1") as f1:
            f1 += [ Task("t{}".format(i)) for i in range(1,8) ]
            f1 += Edit(SLEEP=20)
            f1.t1 += Meter("progress", 1, 100, 90)
            f1.t2 += [ Trigger(["t1"]), Event("a"), Event("b") ]
            f1.t3 += Trigger("t2:a")
            f1.t4 += [ Trigger(["t2"]), Complete("t2:b") ]
            f1.t5 += Trigger("t1:progress ge 30")
            f1.t6 += Trigger("t1:progress ge 60")
            f1.t7 += Trigger("t1:progress ge 90")
        with suite.add_family("f2") as f2:
            f2 += [ Edit(SLEEP=20), [ Task("t{}".format(i)) for i in range(1,6)] ]
            f2.t1 += Time( "00:30 23:30 00:30" )
            f2.t2 += Day( "sunday" )
            f2.t3 += [ Date("1.*.*"), Time("12:00") ]
            f2.t4 += Time("+00:02")
            f2.t5 += Time("00:02")

print(defs)
print("Checking job creation: .ecf -> .job0")
print(defs.check_job_creation())
print("Checking trigger expressions")
assert len(defs.check()) == 0, defs.check()
print("Saving definition to file 'test.def'")
defs.save_as_defs("test.def")

print("Replace suite /test in the server")
defs.test.replace_on_server()

```