



# **CWLAPP: INTEGRATING CWL WITH PARSL**

**- NISHCHAY KARLE**



# Introduction

- CWL (Common Workflow Language) standardizes workflow description
- Parsl enables parallel computing in Python scripts
- CWLApp integrates CWL with Parsl, enhancing automation and scalability

# CWL Overview

- Common Workflow Language (CWL) is an open standard for describing how to run command line tools and connect them to create workflows
- Simplifies writing and sharing complex analysis pipelines
- Facilitates reproducible and scalable scientific workflows

# CommandLineTool

- A vendor-neutral standard for describing the syntax and input/output semantics of command line programs

# CommandLineTool: touch.cwl

```
cwlVersion: v1.0
class: CommandLineTool
baseCommand: touch

inputs:
  filenames:
    type: string[]
    inputBinding:
      position: 1
      separate: true

outputs:
  output_files:
    type: array
    items: File
    outputBinding:
      glob: $(inputs.filenames)
```

```
$ touch file1.txt file2.txt
```

# CommandLineTool: touch.cwl

```
cwlVersion: v1.0
class: CommandLineTool
baseCommand: touch

inputs:
  filenames:
    type: string[]
    inputBinding:
      position: 1
      separate: true

outputs:
  output_files:
    type: array
    items: File
    outputBinding:
      glob: $(inputs.filenames)
```

```
from cwl import CWLApp

touch = CWLApp("touch.cwl")

touch(
    filenames=["file1.txt", "file2.txt"],
    output_files=[
        File("file1.txt"),
        File("file2.txt"),
    ],
).result()
```

```
print(touch.command_template)
```

# CommandLineTool: find.cwl

```
cwlVersion: v1.2
class: CommandLineTool
baseCommand: find

inputs:
  dir:
    type: string
    Default: .
    inputBinding:
      position: 1

  name:
    type: string?
    inputBinding:
      prefix: -name
      separate: true
      position: 2

  maxdepth:
    type: int?
    inputBinding:
      prefix: -maxdepth
      position: 3

outputs:
  stdout:
    type: stdout
  stderr:
    type: stderr
```

```
$ find '.' -name '*.docx' -maxdepth 3
```

# CommandLineTool: find.cwl

```
cwlVersion: v1.2
class: CommandLineTool
baseCommand: find

inputs:
  dir:
    type: string
    Default: .
    inputBinding:
      position: 1

  name:
    type: string?
    inputBinding:
      prefix: -name
      separate: true
      position: 2

  maxdepth:
    type: int?
    inputBinding:
      prefix: -maxdepth
      position: 3

outputs:
  stdout:
    type: stdout
  stderr:
    type: stderr
```

```
find(
  dir="/usr/docs",
  name="*.docx",
  maxdepth=3,
  stdout="find.stdout",
  stderr="find.stderr",
).result()
```

```
$ find '/usr/docs' -name '*.docx' -maxdepth 3
```

```
find(
  name="*.pdf",
  stdout="find.stdout",
  stderr="find.stderr",
).result()
```

```
$ find '.' -name '*.pdf'
```



# Workflows: #1

```
from tools import cat, wc

cat_future = cat(
    from_files=[
        "file1.txt",
        "file2.txt",
        "file3.txt",
    ],
    to_file="combined.txt",
    output_file=File("combined.txt"),
)

wc(
    num_lines=True,
    num_words=True,
    input_files=[cat_future.outputs[0]],
    stdout="wc_stdout.stdout",
    stderr="wc_stderr.stderr",
).result()

with open("wc_stdout.stdout", "r") as f:
    print(f.read())
```

# Workflows: #2

```
from tools import cat

q1_fut = cat(
    from_files=[
        "january_report.csv",
        "february_report.csv",
        "march_report.csv",
    ],
    to_file="q1_report.csv",
    output_file=File("q1_report.csv"),
)

q2_fut = cat(
    from_files=[
        "april_report.csv",
        "may_report.csv",
        "june_report.csv",
    ],
    to_file="q2_report.csv",
    output_file=File("q2_report.csv"),
)

half_year_report = cat(
    from_files=[q1_fut.outputs[0], q2_fut.outputs[0]],
    to_file="half_year_report.csv",
    output_file=File("half_year_report.csv"),
)

half_year_report.result()

with open("half_year_report.csv", "r") as f:
    print(f.read())
```

# Links:

- CWL: <https://www.commonwl.org/>
- CWLApp: <https://github.com/Parsl/cwl-parsl>

**Thank you**