# FCC Dirac Workflows with the Transformation System

André Sailer

CERN-EP-SFT

FCCSW Software Meeting
October 8, 2021

# The input

- trying to implementing workflows as described in https://docs.google.com/document/d/18TAhC62jkE0C5rikOUq9xM-oIL6LpEccLvo4f2HofyY
- Output Folder structure as in google doc: Proposal for a structure for /eos/experiment/fcc/prod (not sure I should share the link here.)

# Workflow 1

Run KKMC followed by Delphes. Translates to these commands (e.g.,)

- `kkmcee --flavour tau --ecms 91.2 --nevts 1000 --outfile events.lhe`
- `DelphesPythia8_EDM4HEP delphes_card_IDEA.tcl edm4hep_output_config.tcl Pythia_LHEinput.cmd outputfile.root`

Output location:

- /eos/experiment/fcc/prod/fcc/ee/91.2gev/tautau/kkmcee/lhef/00012345/
- /eos/experiment/fcc/prod/fcc/ee/91.2gev/tautau/kkmcee/delphes/IDEA/00012345/

Comments:

- Having the input filename hardcoded in `Pythia_LHEinput.cmd` requires effort to have the file events.lhe in the working directory, as the production machinery is all towards creating unique filenames.
    - Copying kkmcee outputfile to events.lhe during the job now
- Input files (delphes card, output config, LHEinput.cmd) stored in “FCCeeConfigPackage”, just some tarball with files stored on the grid, or a folder on CVMFS

# The implementation

- The next couple slides of code are not something that most people will ever have to look at
- illustrate some of the parameters being used,
- There is a lot of code being glossed over and not shown
- In principle any workflow is possible, but to bring consistency some things are rather rigid

# Configuring KKMC

```
# self, task come from outside
kkmcee = KKMC()
kkmcee.setVersion(self.generatorVersion)
# from tautau to tau
kkmcee.setEvtType(''.join(task.meta['EvtType'][:len(task.meta['EvtType']) / 2]).capitalize())
kkmcee.setNumberOfEvents(task.eventsPerJob)
kkmcee.setEnergy(task.meta['Energy'])
self._setApplicationOptions('KKMC', kkmcee, task.applicationOptions)
task.datatype = 'lhef'
task.generator = 'kkmcee'
```

# Configuring the Delphes App

```
ga = GaudiApp()  # oh the abuse
ga.setVersion('key4hep-latest')
ga.setExecutable("DelphesPythia8_EDM4HEP")
ga.setInputFileFlag('')
ga.setOutputFileFlag("delphes_card_IDEA.tcl edm4hep_output_config.tcl Pythia_LHEinput.cmd")
ga.setNumberOfEvents(task.eventsPerJob)
ga.setEnergy(task.meta['Energy'])
ga.detector = task.detector
ga._extension = 'root'
ga.datatype = 'delphes'
```

# Creating the ProductionJob

```
genProd = self.getProductionJob()
genProd.setConfigPackage(appName=self.configPackage, version=self.configVersion)
genProd.evttype = task.meta['EvtType']
genProd.setProdType('MCGeneration')
genProd.setWorkflowName(prodName)
genProd.append(kkmcee)
genProd.append(ga)
genProd.setNbOfTasks(task.nbTasks)
genProd.createProduction()
```

# Setting output folder structure and metadata

Accelerator, initial state, energy, event final state, generator, fileformat, detector model, prodid

```
currentPathList = [basepath, energypath, evttypepath]
path = os.path.join(*currentPathList)
self.finalMetaDict[path] = {"EvtType": self.evttype}
currentPathList.append(self.generator)
path = os.path.join(*currentPathList)
self.finalMetaDict[path] = {'Generator': self.generator}
path = os.path.join(path, application.datatype)
self.finalMetaDict[path] = {'Datatype': application.datatype}
if hasattr(application, "detectortype") and application.detectortype:
  path = os.path.join(path, application.detectortype)
  self.finalMetaDict[path] = {"DetectorType": application.detectortype}
```

# Output Structure, Metadata

```
/fcc/ee/91.2gev/: {'Energy': '91.2'}
/fcc/ee/91.2gev/tautau/: {'EvtType': 'tautau'}
/fcc/ee/91.2gev/tautau/kkmcee: {'Generator': 'kkmcee'}
/fcc/ee/91.2gev/tautau/kkmcee/delphes: {'Datatype': 'delphes'}
/fcc/ee/91.2gev/tautau/kkmcee/delphes/IDEA: {'DetectorType': 'IDEA'}
/fcc/ee/91.2gev/tautau/kkmcee/delphes/IDEA/00012345: {'NumberOfEvents': 1000, 'ProdID': 12345}
/fcc/ee/91.2gev/tautau/kkmcee/lhef: {'Datatype': 'lhef'}
/fcc/ee/91.2gev/tautau/kkmcee/lhef/00012345: {'NumberOfEvents': 1000, 'ProdID': 12345}
Setting non searchable metadata information:
  /fcc/ee/91.2gev/tautau/kkmcee/delphes/IDEA/00401242:
  {'SWPackages': 'kkmc.key4hep_nightly;gaudiapp.key4hep-latest'}
Setting non searchable metadata information:
  /fcc/ee/91.2gev/tautau/kkmcee/lhef/00401242:
  {'SWPackages': 'kkmc.key4hep_nightly;gaudiapp.key4hep-latest'}
  # that last one should get fixed, as output doesn't depend on gaudiapp version
```

Filenames

```
  tautau_lhef_12345_<N>.lhe
  tautau_delphes_12345_<N>.root
```

# How to create the transformation

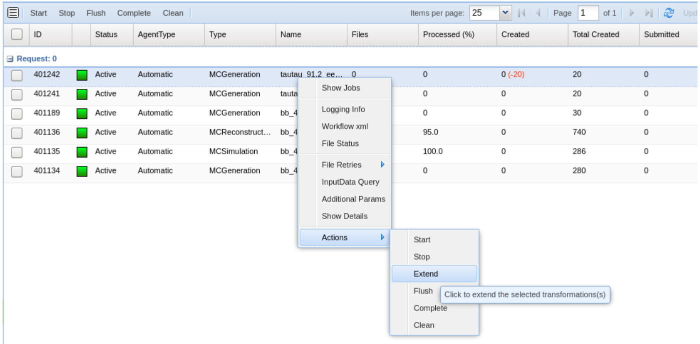
- `source /cvmfs/clicdp.cern.ch/DIRACpreview/bashrc` # pre-prod instance
- `dirac-proxy-init -g fcc_prod` # for the privileged
- `dirac-fcc-make-productions -p > configFile` # modify
- `dirac-fcc-make-productions -f configFile`

# A Config File

```
# [snip]
configVersion = key4hep-devel
configPackage = fccConfig
eventsPerJobs = 1000

# can define list of energies or processes ("scalar" product)
energies = 91.2, 150, 300
processes = tautau, mumu, ee

productionLogLevel = VERBOSE
outputSE = CERN-DST-EOS

# different workflows as differnt prod types
ProdTypes = Gen

# [snip]
```

# Need more files

```
https://voilcdiracwebapp2.cern.ch/DIRAC/s:ILC-Development/g:fcc_prod/?view=tabs&theme=Crisp&url_state=1|*DIRAC.TransformationMonitor.classes.TransformationMonitor:,
```



Request: 0

| □ | ID | | Status | AgentType | Type | Name | Files | Processed (%) | Created | Total Created | Submitted |
|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | 401242 | 🟩 | Active | Automatic | MCGeneration | tautau_91.2_ee... | 0 | 0 | 0 (-20) | 20 | 0 |
| □ | 401241 | 🟩 | Active | Automatic | MCGeneration | tauta | | 0 | 0 | 20 | 0 |
| □ | 401189 | 🟩 | Active | Automatic | MCGeneration | bb_4 | | 0 | 0 | 30 | 0 |
| □ | 401136 | 🟩 | Active | Automatic | MCReconstruct... | bb_4 | | 95.0 | 0 | 740 | 0 |
| □ | 401135 | 🟩 | Active | Automatic | MCSimulation | bb_4 | | 100.0 | 0 | 286 | 0 |
| □ | 401134 | 🟩 | Active | Automatic | MCGeneration | bb_4 | | 0 | 0 | 280 | 0 |

or

```
dirac-ilc-add-tasks-to-prod.py 401242 30
```

# Find the files, production info

```
dirac-dms-find-lfns Path=/ ProdID=401242 Datatype=delphes
dirac-dms-find-lfns Path=/ EvtType=tautau Energy=91.2 Datatype=delphes \
      DetectorType=IDEA
dirac-ilc-get-info -p 401242
dirac-ilc-get-prod-log -P 401242
```

# Next steps

- Implement more workflows
  - Need to know what exact commands to run, and where the config files are
- Improve configurability for workflows
- Release dirac-fcc-make-production command to production instance