LA-UR-21-23116
Approved for public release; distribution is unlimited.

Title:	BEE-FY21 P6-1: Archive, clone, and re-run workflows 2.3.6.01 - LANL ATDM ST / STNS01-21 Milestone Completion Documentation
Author(s):	Randles, Timothy C.
Intended for:	ECP Project Milestone completion documentation
Issued:	2021-04-01

Los Alamos
NATIONAL LABORATORY

BEE-FY21 P6-1: Archive, clone, and re-run workflows

2.3.6.01 – LANL ATDM ST / STNS01-21 Milestone Completion Documentation

Tim Randles
for the BEE Team

March 31, 2021

Activity Description

Archive, clone, and re-run workflows

BEE will have the ability to archive, clone, and re-run workflows that have been previously executed. This will enable reproducibility and portability of scientific workflows both within and between DOE facilities.

Execution Plan

To complete this activity, the BEE project team will:

- add new functionality to the BEE client and BEEWorkflowManager to produce an archive of an existing workflow
- add new functionality to the BEE client to copy an archived workflow
- add new functionality to the BEE client and BEEWorkflowManager to examine and reset workflow state data so that a workflow can be re-executed

Completion Criteria

When this activity is complete, BEE will:

1. provide the user with a mechanism to archive a workflow
2. provide the user with the ability to "clone" or "copy" a workflow
3. provide the user with the ability to reset any state data in a workflow and then re-execute the workflow

Completion Criteria

1. provide the user with a mechanism to archive a workflow

```
# BEE CONFIGURATION FILE #
[DEFAULT]
bee_workdir = /yellow/users/trandles/.beeflow
workload_scheduler = Slurm
use_archive = True
```

User configures BEE to archive workflows

```
(beeflow-1DYj9qyG-py3.6) [trandles@fg-fey1 temp]$ python3 ../client.py
Welcome to BEE Client!
0) Submit Workflow
1) List Workflows
2) Start Workflow
3) Query Workflow
4) Pause Workflow
5) Resume Workflow
6) Cancel Workflow
7) Copy Workflow
8) ReExecute Workflow
9) Exit
$ 0
What will be the name of the job?
$ timtest
What is the workflow path?
$ cr.cwl
Job submitted! Your workflow id is fc00925d-dd9a-4019-9ebf-54ad69c9e00a.
```

```
(beeflow-1DYj9qyG-py3.6) [trandles@fg-fey1 temp]$ python3 ../client.py
Welcome to BEE Client!
0) Submit Workflow
1) List Workflows
2) Start Workflow
3) Query Workflow
4) Pause Workflow
5) Resume Workflow
6) Cancel Workflow
7) Copy Workflow
8) ReExecute Workflow
9) Exit
$ 1
Name    ID                                      Status
timtest fc00925d-dd9a-4019-9ebf-54ad69c9e00a    Archived
(beeflow-1DYj9qyG-py3.6) [trandles@fg-fey1 temp]$ ls ~/.beeflow/archives/
fc00925d-dd9a-4019-9ebf-54ad69c9e00a.tgz
```

When a user submits a workflow a unique workflow ID is created. After the workflow finishes running, the workflow archive is automatically created in an archives directory.

2. provide the user with the ability to ”clone” or “copy” a workflow

```
Name    ID                                      Status
timtest fc00925d-dd9a-4019-9ebf-54ad69c9e00a    Archived
(beeflow-1DYj9qyG-py3.6) [trandles@fg-fey1 temp]$ ls ~/.beeflow/archives/
fc00925d-dd9a-4019-9ebf-54ad69c9e00a.tgz
```

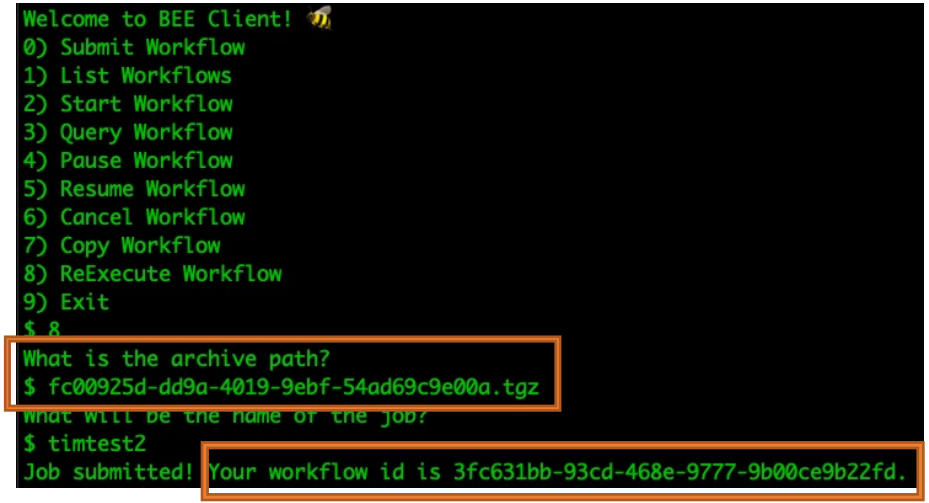
In the screenshot above user “trandles” on host “fg-fey” has an archived workflow.

In the screenshot on the right user “tcr” on host “case” runs the BEE client, queries the workflow manager on fg-fey for a list of workflows, and copies that workflow to case.

This was accomplished by using an SSH tunnel to carry the BEE traffic between case and fg-fey, but it demonstrates the ability of BEE to enable a user to copy their workflows between systems.

```
(beeflow-71jFR0ii-py3.8) [tcr@case CWL]$ ls
(beeflow-71jFR0ii-py3.8) [tcr@case CWL]$ python3 ~/BEE_Private/src/beeflow/client/client.py
Welcome to BEE Client!
0) Submit Workflow
1) List Workflows
2) Start Workflow
3) Query Workflow
4) Pause Workflow
5) Resume Workflow
6) Cancel Workflow
7) Copy Workflow
8) ReExecute Workflow
9) Exit
$ 1
Name    ID                                      Status
timtest fc00925d-dd9a-4019-9ebf-54ad69c9e00a    Archived
(beeflow-71jFR0ii-py3.8) [tcr@case CWL]$ python3 ~/BEE_Private/src/beeflow/client/client.py
Welcome to BEE Client!
0) Submit Workflow
1) List Workflows
2) Start Workflow
3) Query Workflow
4) Pause Workflow
5) Resume Workflow
6) Cancel Workflow
7) Copy Workflow
8) ReExecute Workflow
9) Exit
$ 7
What is the workflow id?
$ fc00925d-dd9a-4019-9ebf-54ad69c9e00a
Where do you want to save it?
$ .
(beeflow-71jFR0ii-py3.8) [tcr@case CWL]$ ls
fc00925d-dd9a-4019-9ebf-54ad69c9e00a.tgz
```

3. provide the user with the ability to reset any state data in a workflow and then re-execute the workflow



To re-execute a workflow, the user passes the BEE client an archived workflow. This archived workflow is ingested by BEE and produces a new workflow id for the new run.

3. provide the user with the ability to reset any state data in a workflow and then re-execute the workflow


```
{"outputs":["clamr/outfile"],"name":"clamr","workflow_id":"fc00925d-dd9a-4019-9ebf-54ad69c9e00a"
,"task_id":"70f7500c-f991-4db2-a3ae-43ff2dc209bb","inputs":["infile"],"command":"/clamr/CLAMR-ma
ster/clamr_cpuonly -n 32 -l 3 -t 5000 -i 10 -g 25 -G png"}
```

```
{"container_runtime":"Charliecloud","host":"fg-fey1.lanl.gov","state":"RUNNING","container_hash"
:"e07734d470200bd8bebc311d9456370f","job_id":416333}
```


In the screenshot above, taken from the neo4j graph database, we see the original workflow id and a slurm job id when the workflow was executing. In the screenshot below we see the new workflow id and a different slurm job id from the new run of the same workflow.


```
{"outputs":["ffmpeg/outfile"],"name":"ffmpeg","workflow_id":"3fc631bb-93cd-468e-9777-9b00ce9b22f
d","task_id":"1d2c91b1-d8a7-48cc-9020-f92cb143b5f0","inputs":["clamr/outfile"],"command":"ffmpeg
 -y -f image2 -i $HOME/graphics_output/graph%05d.png -r 12 -s 800×800 -pix_fmt yuv420p $HOME/CLA
MR_movie.mp4"}
```

```
{"container_runtime":"Charliecloud","host":"fg-fey1.lanl.gov","state":"RUNNING","container_hash"
:"e07734d470200bd8bebc311d9456370f","job_id":416342}
```

3. provide the user with the ability to reset any state data in a workflow and then re-execute the workflow

```
Welcome to BEE Client!
0) Submit Workflow
1) List Workflows
2) Start Workflow
3) Query Workflow
4) Pause Workflow
5) Resume Workflow
6) Cancel Workflow
7) Copy Workflow
8) ReExecute Workflow
9) Exit
$ 1
Name     ID                                      Status
timtest  fc00925d-dd9a-4019-9ebf-54ad69c9e00a    Archived
timtest2         3fc631bb-93cd-468e-9777-9b00ce9b22fd    Archived
```

Listing workflows now shows two archives, the original workflow run and the re-executed workflow.