

# Submitted Workflows Troubleshooting

In this article, we create an error in a workflow to guide you through the process of troubleshooting a workflow that you've submitted to Treasure Data.

- [Prerequisites](#)
  - [Introductory Tutorial](#)
- [Create an error to debug](#)
- [Troubleshooting](#)
  - [Determine what tasks failed](#)
  - [Review logs of the failed task](#)
  - [Fix the query](#)
  - [Push the fix to Treasure Data](#)
  - [Retry the workflow session](#)

## Prerequisites

### Introductory Tutorial

If you haven't already, start by going through the [TD Workflows Introductory Tutorial](#).

You will download and use the workflow project in the tutorial.

## Create an error to debug

Navigate to the `nasdaq\_analysis` directory from the introductory tutorial.

Use the following syntax to create an error for us to debug:

```
SELECT TD_DATE_TRUNC('month', time), AVG(daily_avg_open) AS monthly_avg_open, AVG(daily_avg_close) AS
month_avg_close
FROM daily_open
GROUP BY 1
EOF
```

Push the broken workflow to Treasure Data

```
$ td wf push nasdaq_analysis
# Submitting workflow "nasdaq_analysis"...
# Done!
```

Start the workflow, on Treasure Data's side:

```
$ td wf start nasdaq_analysis nasdaq_analysis --session now
```

Check failure status:

```
$ td wf session nasdaq_analysis nasdaq_analysis
```

You should see the following as your output

```
2016-05-11 16:40:24 +0900: Digdag v0.6.1
Session attempts:
  attempt id: 100
  uuid: ef704elf-3eb5-4ba7-9be0-4ebfaeee4424
  project: nasdaq_analysis
  workflow: nasdaq_analysis
  session time: 2016-05-11 07:38:15 +0000
  retry attempt name:
  params: {"td":{"apikey":"..."},"last_session_time":"2016-05-11T00:00:00+00:00","next_session_time":"2016-05-12T00:00:00+00:00"}
  created at: 2016-05-11 16:38:17 +0900
  kill requested: false
  status: error
```

## Troubleshooting

### Determine what tasks failed

In above example, `attempt_id = 100`.

```
$ td wf tasks <attempt_id>
```

The command should return output similar to the following:

```
2016-05-16 21:18:19 -0700: Digdag v0.7.1
  id: 1105
  name: +nasdaq_analysis
  state: group_error
  config: {"schedule":{"daily>":"07:00:00"},"_export":{"td":{"database":"workflow_temp"}}}
  parent: null
  upstreams: []
  export params: {"td":{"database":"workflow_temp"}}
  store params: {}
  state params: {}

  id: 1106
  name: +nasdaq_analysis+task1
  state: success
  config: {"td>":"queries/daily_open.sql","create_table":"daily_open"}
  parent: 1105
  upstreams: []
  export params: {}
  store params: {"td":{"last_job_id":"66338029"}}
  state params: {}

  id: 1107
  name: +nasdaq_analysis+task2
  state: error
  config: {"td>":"queries/monthly_open.sql","create_table":"monthly_open"}
  parent: 1105
  upstreams: [1106]
  export params: {}
  store params: {}
  state params: {}
```

You can see under the last task listed, named `+nasdaq_analysis+task2` that **state: error**, meaning this task is the one that failed.

### Review logs of the failed task

The command to get the logs for a particular tasks is as follows:

```
$ td wf logs <attempt_id> <task_name>
```

Specifically, put the following:

```
$ td wf logs <attempt_id> +nasdaq_analysis+task2
```

Review the output to determine the cause of the errors.

- You can also use the job id to review error logs in TD Console.

## Fix the query

Fix the query and rerun the workflow.

```
$ cat > queries/monthly_open.sql <<EOF
SELECT TD_DATE_TRUNC('month', time), AVG(daily_avg_open) AS
monthly_avg_open, AVG(daily_avg_close) AS month_avg_close
FROM daily_open
GROUP BY 1
EOF
```

## Push the fix to Treasure Data

```
$ td wf push nasdaq_analysis
```

## Retry the workflow session

Rerun the workflow.

```
$ td wf retry <attempt_id> --name fix-typo --latest-revision --all
```

Quickly run `td wf attempts` to see the new session attempt running. Run it again, and you'll likely see it succeeded successfully.

The most recent attempt has the same `session time` as the previous attempt that failed. This is the benefit of using `retry` in this instance, instead of `start`. This is particularly important if you have a daily scheduled workflow, and you only want to retry the current day's session using any time-related parameters embedded into the workflow.

Alternatively, you can use `--resume` to only rerun starting at the failed task and all subsequent tasks.