

Harmony Documentation

September 17, 2018

1 sample_inputs

1.1 rgt.input.master

This file contains an example input file for starting tests. Harmony parses this file to get:

- **path_to_tests:** The path to where tests are stored.
Example: `path_to_tests = /usr/guest/home/sample_run`
- **test:** A test containing the program name and the test name.
Example: `test = GTC4 test_0001node`

1.2 sample_run/GTC4/test_0001node/

Directory containing example code for what a test might look like. `/Run_Archive/123456789.0123/stdout.txt` contains the output from the test while running. The numbers are unique identifiers pertaining to that specific job. `Scripts/` contains files necessary for running the test. `Status/rgt_status.txt` contains start time, uniqueID, jobID, and the status for building, submitting, and running. Subdirectories of the `Status/` directory contain information pertaining to how the job is running. `job_id.txt` contains the job's id on LSF and `job_status.txt` contains the jobs current status.

2 scripts

2.1 connect.py

Contains method for connecting to lsf.

Example:

```
from connect import connect
connect.connect()
```

2.2 job_status.py

Check the status of any set of jobs.

```
Example: from job_status import JobStatus
          js = JobStatus(queue='batch')
          jobs = js.get_jobs(user="guest", job_status=["pending",
"running"])
```

2.3 parse_file.py

This file parses text files. Wow. Not self explanatory at all! It can parse the job_id.txt file from a test directory and can also parse an rgt.input file with some nice error throwing.

```
Example: from parse_file import ParseJobID
          id = ParseJobID.pars_file("/usr/guest/home/sample_run/GTC4/
test_0001node/Status/latest/job_id.txt")

          from parse_file import ParseRGInput
          path_to_tests, test_list = ParseRGInput.parse_file(
"/usr/guest/home/rgt.input.master")
```

2.4 test_status.py

This file checks whether each test from an rgt.input file is running and will return a string for notification.

```
Example: from test_status import check_tests
          error = check_tests("/path/to/rgt.input")
```

2.5 notifications

2.5.1 email/email_send.py

This method will send an email using SMTP through the localhost.

```
Example: from email_send import send_message
          send_message(text_doc="/path/to/email.txt",
          subject="Notification", sender="OLCF-notice",
          reciever="guest@example.com")
```

3 Slack

We can use either email or slack to notify people. When using Slack there are a couple ways to run such an application.

3.1 Flask

Flask a very simple web framework usable through python. It is a "Non Full-Stack" framework meaning that it is nice and simple to use but does not have all the capabilities that a full-stack framework like Django might have. A full-stack framework contains everything from the data storage up to the user interface. Since Slack is our interface, a nice simple library like Flask does the trick!

3.2 Running

At the moment, the server is started as a process and sits open continuously. We can offload this to a server but trying to interface with Summit from a separate server while authenticating through Slack seems like trouble indeed. An example of this might be running it on AWS. The server needs to be continuously up so that it can receive and initiate requests whenever. I have no clue about how to secure any such beast. Currently it appears as though the server must always be up and running to use the application and unless we get a permanent host address, the app will need to be reinstalled whenever the server restarts.

It can be run using the following command with the correct ids.

Example: `SLACK_VERIFICATION_TOKEN=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
SLACK_TEAM_ID=xxxxxxx FLASK_APP=path/to/flask_app.py flask run`

To watch the server and make sure it is doing ok just use `ngrok http port_num` where `port_num` is the what port the server is running from (default: 5000).