
Swift log processing tools

Source control `$LastChangedRevision$`

Table of Contents

1. Overview	1
2. Web page about a run	1
3. CEDPS logs	2
4. Event/transition channels	2

1. Overview

There is an under-development package of Swift log processing utilities.

This code is not distributed in the main Swift release. Instead, it must be obtained from the Swift SVN repository, in the directory `log-processing/`:

```
svn co https://svn.ci.uchicago.edu/svn/vdl2/log-processing
```

2. Web page about a run

```
make LOG=/path/to/readData-20080304-0903-xgqf5nhe.log clean webpage
```

This will create a web page in the `log-processing` directory giving information about the run that is extracted from the log file. If the above command is used before a run is completed, the web page will report information about the workflow progress so far.

Additional more expensive to compute information can added to this using the following makefile targets, which should be added to the command line between the `clean` and makefile targets.

`webpage.kara` - more detailed information about internal Karajan-level execution

`webpage.clusters` - details of job clustering. This will fail if no job clustering occurred.

`webpage.falkon` - details of falkon workers. This will fail if no Falkon log file is present.

`webpage.weights` - details of site scoring

`webpage.info` - details of `execute-site` wrapper logs. For versions of Swift prior to r1700, you will need to stage the `*-info` logs back to the same place as the Swift log manually. After r1700, they are staged back automatically under the control of the `wrapperlog.always.transfer` property. The `IDIR` variable must be set to point to the directory containing the logs:

```
make LOG=/path/fmri-20080304-0901-h8h78lnf.log \  
      IDIR=/path/130-fmri-20080304-0901-h8h78lnf.d/ clean webpage.info webpage
```

webpage.kickstart - similar to the webpage.info target, this gives execute-site kickstart record information.

3. CEDPS logs

The log processing tools can output transition streams in CEDPS logging format:

```
make LOG=/path/to/readData-20080304-0903-xgqf5nhe.log clean execute.cedps
```

4. Event/transition channels

Various event channels are extracted from the log files and made available as `.event` and `.transition` files. These roughly correspond to processes within the Swift runtime environment.

These streams are then used to provide the data for the various output formats, such as graphs, web pages and CEDPS log format.

The available basic stream names are: `execute`, `execute2`, `kickstart`, `info`, `karajan`, `clusters`, `stageout`, `stagein`, `work-flow`