Savanna API

Generated by Doxygen 1.8.13

Contents

1 Namespace Index **1**

1.1 Namespace List 1

2 Hierarchical Index **3**

2.1 Class Hierarchy 3

3 Class Index **5**

3.1 Class List 5

4 File Index **7**

4.1 File List 7

5 Namespace Documentation **9**

5.1 codar Namespace Reference 9

5.2 codar.savanna Namespace Reference 9

5.2.1 Detailed Description 9

5.3 codar.savanna.consumer Namespace Reference 10

5.3.1 Detailed Description 10

5.4 codar.savanna.exc Namespace Reference 10

5.4.1 Detailed Description 10

5.5 codar.savanna.machines Namespace Reference 10

5.5.1 Detailed Description 11

5.5.2 Function Documentation 11

5.5.2.1 get_by_name() 11

5.5.3 Variable Documentation 11

5.5.3.1 cori 11
5.5.3.2 local 11
5.5.3.3 SCHEDULER_OPTIONS 12
5.5.3.4 summit 12
5.5.3.5 theta 12
5.5.3.6 titan 12
5.6 codar.savanna.main Namespace Reference 13
5.6.1 Detailed Description 13
5.6.2 Function Documentation 13
5.6.2.1 get_job_id() 13
5.6.2.2 main() 13
5.6.2.3 parse_args() 13
5.6.3 Variable Documentation 13
5.6.3.1 consumer 14
5.7 codar.savanna.model Namespace Reference 14
5.7.1 Detailed Description 14
5.7.2 Variable Documentation 14
5.7.2.1 KILL_WAIT 14
5.7.2.2 RETURN_NAME 15
5.7.2.3 STDERR_NAME 15
5.7.2.4 STDOUT_NAME 15
5.7.2.5 WAIT_DELAY_GIVE_UP 15
5.7.2.6 WAIT_DELAY_KILL 15
5.7.2.7 WALLTIME_NAME 15
5.8 codar.savanna.node_layout Namespace Reference 16
5.9 codar.savanna.producer Namespace Reference 16
5.9.1 Detailed Description 16
5.10 codar.savanna.runners Namespace Reference 16
5.10.1 Variable Documentation 16
5.10.1.1 aprun 16

5.10.1.2 jsrun 17

5.10.1.3 mpiexec 17

5.10.1.4 srun 17

5.11 codar.savanna.scheduler Namespace Reference 17

5.11.1 Detailed Description 17

5.12 codar.savanna.status Namespace Reference 17

5.12.1 Detailed Description 18

5.12.2 Variable Documentation 18

5.12.2.1 DONE 18

5.12.2.2 KILLED 18

5.12.2.3 NOT_STARTED 18

5.12.2.4 REASON_EXCEPTION 19

5.12.2.5 REASON_FAILED 19

5.12.2.6 REASON_NOFIT 19

5.12.2.7 REASON_SUCCEEDED 19

5.12.2.8 REASON_TIMEOUT 19

5.12.2.9 RUNNING 19

5.13 codar.savanna.summit_helper Namespace Reference 20

5.13.1 Function Documentation 20

5.13.1.1 create_erf_file() 20

5.13.1.2 get_nodes_reqd() 20

6 Class Documentation **21**

6.1 codar.savanna.scheduler.JobList Class Reference 21

6.1.1 Detailed Description 22

6.1.2 Constructor & Destructor Documentation 22

6.1.2.1 __init__() 22

6.1.3 Member Function Documentation 22

6.1.3.1 __len__() 22

6.1.3.2 add_job() 23

6.1.3.3 pop_job() 23

6.2 codar.savanna.producer.JSONFilePipelineReader Class Reference 23

6.2.1 Detailed Description 24

6.2.2 Constructor & Destructor Documentation 24

6.2.2.1 __init__() 24

6.2.3 Member Function Documentation 25

6.2.3.1 read_pipelines() 25

6.2.4 Member Data Documentation 25

6.2.4.1 file_path 25

6.3 codar.savanna.machines.Machine Class Reference 25

6.3.1 Detailed Description 26

6.3.2 Constructor & Destructor Documentation 26

6.3.2.1 __init__() 27

6.3.3 Member Function Documentation 27

6.3.3.1 get_nodes_reqd() 27

6.3.3.2 get_scheduler_options() 27

6.3.4 Member Data Documentation 27

6.3.4.1 dataspaces_servers_per_node 27

6.3.4.2 name 28

6.3.4.3 node_class 28

6.3.4.4 node_exclusive 28

6.3.4.5 processes_per_node 28

6.3.4.6 runner_name 28
6.3.4.7 scheduler_name 28
6.3.4.8 scheduler_options 29
6.4 codar.savanna.machines.MachineNode Class Reference 29
6.4.1 Detailed Description 29
6.4.2 Constructor & Destructor Documentation 29
6.4.2.1 __init__() 30
6.4.3 Member Function Documentation 30
6.4.3.1 to_json() 30
6.4.3.2 validate_layout() 30
6.4.4 Member Data Documentation 30
6.4.4.1 cpu 30
6.4.4.2 gpu 30
6.5 codar.savanna.exc.MachineNotFound Class Reference 31
6.5.1 Detailed Description 31
6.5.2 Constructor & Destructor Documentation 32
6.5.2.1 __init__() 32
6.6 codar.savanna.runners.MPIRunner Class Reference 32
6.6.1 Detailed Description 33
6.6.2 Constructor & Destructor Documentation 33
6.6.2.1 __init__() 33
6.6.3 Member Function Documentation 34
6.6.3.1 wrap() 34
6.6.4 Member Data Documentation 34
6.6.4.1 exe 34
6.6.4.2 hostfile 34
6.6.4.3 nodes_arg 34
6.6.4.4 nprocs_arg 34
6.6.4.5 tasks_per_node_arg 35
6.7 codar.savanna.model.NodeConfig Class Reference 35

6.7.1 Detailed Description 35
6.7.2 Constructor & Destructor Documentation 35
6.7.2.1 __init__() 35
6.7.3 Member Data Documentation 35
6.7.3.1 cpu 36
6.7.3.2 gpu 36
6.7.3.3 num_ranks_per_node 36
6.8 codar.savanna.node_layout.NodeLayout Class Reference 36
6.8.1 Detailed Description 37
6.8.2 Constructor & Destructor Documentation 38
6.8.2.1 __init__() 38
6.8.3 Member Function Documentation 38
6.8.3.1 add_node() 38
6.8.3.2 as_data_list() 38
6.8.3.3 codes_per_node() 38
6.8.3.4 copy() 39
6.8.3.5 default_no_share_layout() 39
6.8.3.6 get_node_containing_code() 39
6.8.3.7 group_codes_by_node() 39
6.8.3.8 populate_remaining() 40
6.8.3.9 ppn() 40
6.8.3.10 serialize_to_dict() 40
6.8.3.11 shared_nodes() 40
6.8.3.12 validate() 40
6.8.4 Member Data Documentation 41
6.8.4.1 layout_list 41
6.8.4.2 layout_map 41
6.9 codar.savanna.model.Pipeline Class Reference 41
6.9.1 Detailed Description 42
6.9.2 Constructor & Destructor Documentation 42

6.9.2.1 __init__() 43
6.9.3 Member Function Documentation 43
6.9.3.1 add_done_callback() 43
6.9.3.2 add_fatal_callback() 43
6.9.3.3 force_kill_all() 43
6.9.3.4 from_data() 44
6.9.3.5 get_nodes_used() 44
6.9.3.6 get_pids() 44
6.9.3.7 get_state() 44
6.9.3.8 join_all() 44
6.9.3.9 remove_done_callback() 45
6.9.3.10 remove_fatal_callback() 45
6.9.3.11 run_finished() 45
6.9.3.12 run_post_process_script() 45
6.9.3.13 set_ppn() 45
6.9.3.14 set_total_nodes() 46
6.9.3.15 start() 46
6.9.4 Member Data Documentation 46
6.9.4.1 done_callbacks 46
6.9.4.2 fatal_callbacks 46
6.9.4.3 id 46
6.9.4.4 kill_on_partial_failure 47
6.9.4.5 launch_mode 47
6.9.4.6 log_prefix 47
6.9.4.7 machine_name 47
6.9.4.8 node_layout 47
6.9.4.9 nodes_assigned 47
6.9.4.10 post_process_args 48
6.9.4.11 post_process_script 48
6.9.4.12 post_process_stop_on_failure 48

6.9.4.13 runs 48
6.9.4.14 total_nodes 48
6.9.4.15 total_procs 48
6.9.4.16 working_dir 49
6.10 codar.savanna.consumer.PipelineRunner Class Reference 49
6.10.1 Detailed Description 50
6.10.2 Constructor & Destructor Documentation 50
6.10.2.1 __init__() 50
6.10.3 Member Function Documentation 50
6.10.3.1 add_pipeline() 51
6.10.3.2 kill_all() 51
6.10.3.3 pipeline_fatal() 51
6.10.3.4 pipeline_finished() 51
6.10.3.5 run_finished() 52
6.10.3.6 run_pipelines() 52
6.10.3.7 stop() 52
6.10.4 Member Data Documentation 52
6.10.4.1 allocated_nodes 52
6.10.4.2 free_cv 53
6.10.4.3 free_nodes 53
6.10.4.4 job_list 53
6.10.4.5 job_list_cv 53
6.10.4.6 machine_name 53
6.10.4.7 max_nodes 53
6.10.4.8 pipelines 54
6.10.4.9 pipelines_lock 54
6.10.4.10 ppn 54
6.10.4.11 runner 54
6.11 codar.savanna.status.PipelineState Class Reference 55
6.11.1 Detailed Description 55

6.11.2 Constructor & Destructor Documentation 56
6.11.2.1 __init__() 56
6.11.3 Member Function Documentation 56
6.11.3.1 as_data() 56
6.11.4 Member Data Documentation 56
6.11.4.1 id 56
6.11.4.2 reason 56
6.11.4.3 return_codes 57
6.11.4.4 state 57
6.12 codar.savanna.model.Run Class Reference 57
6.12.1 Detailed Description 59
6.12.2 Constructor & Destructor Documentation 59
6.12.2.1 __init__() 59
6.12.3 Member Function Documentation 59
6.12.3.1 add_callback() 59
6.12.3.2 close() 60
6.12.3.3 create_node_config() 60
6.12.3.4 exception() 60
6.12.3.5 from_data() 60
6.12.3.6 get_nodes_used() 61
6.12.3.7 get_pid() 61
6.12.3.8 get_returncode() 61
6.12.3.9 join() 61
6.12.3.10 kill() 61
6.12.3.11 killed() 62
6.12.3.12 mpmd_run() 62
6.12.3.13 remove_callback() 62
6.12.3.14 run() 62
6.12.3.15 set_runner() 62
6.12.3.16 succeeded() 63

6.12.3.17 timed_out() 63
6.12.4 Member Data Documentation 63
6.12.4.1 args 63
6.12.4.2 callbacks 63
6.12.4.3 depends_on_runs 63
6.12.4.4 env 64
6.12.4.5 erf_file 64
6.12.4.6 exe 64
6.12.4.7 hostfile 64
6.12.4.8 log_prefix 64
6.12.4.9 machine 64
6.12.4.10 name 65
6.12.4.11 node_config 65
6.12.4.12 nodes 65
6.12.4.13 nodes_assigned 65
6.12.4.14 nprocs 65
6.12.4.15 res_set 65
6.12.4.16 return_path 66
6.12.4.17 runner 66
6.12.4.18 runner_override 66
6.12.4.19 sched_args 66
6.12.4.20 sleep_after 66
6.12.4.21 stderr_path 66
6.12.4.22 stdout_path 67
6.12.4.23 tasks_per_node 67
6.12.4.24 timeout 67
6.12.4.25 walltime_path 67
6.12.4.26 working_dir 67
6.13 codar.savanna.runners.Runner Class Reference 68
6.13.1 Detailed Description 68

6.13.2 Member Function Documentation 68
6.13.2.1 wrap() 69
6.14 codar.savanna.exc.SavannaException Class Reference 69
6.14.1 Detailed Description 70
6.15 codar.savanna.machines.SummitNode Class Reference 70
6.15.1 Detailed Description 71
6.15.2 Constructor & Destructor Documentation 71
6.15.2.1 __init__() 71
6.15.3 Member Function Documentation 71
6.15.3.1 to_json() 71
6.15.3.2 validate_layout() 71
6.16 codar.savanna.runners.SummitRunner Class Reference 72
6.16.1 Detailed Description 73
6.16.2 Constructor & Destructor Documentation 73
6.16.2.1 __init__() 73
6.16.3 Member Function Documentation 73
6.16.3.1 wrap() 73
6.16.3.2 wrap_deprecated() 74
6.16.4 Member Data Documentation 74
6.16.4.1 bind_arg 74
6.16.4.2 cpus_per_rs_arg 74
6.16.4.3 exe 74
6.16.4.4 gpus_per_rs_arg 74
6.16.4.5 launch_distribution_arg 75
6.16.4.6 machine 75
6.16.4.7 nrs_arg 75
6.16.4.8 rs_per_host_arg 75
6.16.4.9 tasks_per_rs_arg 75
6.17 codar.savanna.status.WorkflowStatus Class Reference 76
6.17.1 Detailed Description 76
6.17.2 Constructor & Destructor Documentation 77
6.17.2.1 __init__() 77
6.17.3 Member Function Documentation 77
6.17.3.1 set_state() 77
6.17.4 Member Data Documentation 77
6.17.4.1 file_path 77

7 File Documentation **79**

7.1 __init__.py File Reference 79

7.2 consumer.py File Reference 79

7.3 exc.py File Reference 79

7.4 machines.py File Reference 80

7.5 main.py File Reference 80

7.6 model.py File Reference 80

7.7 node_layout.py File Reference 81

7.8 producer.py File Reference 81

7.9 runners.py File Reference 81

7.10 scheduler.py File Reference 82

7.11 status.py File Reference 82

7.12 summit_helper.py File Reference 82

Index **83**

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

codar 9
codar.savanna 9
codar.savanna.consumer 10
codar.savanna.exc 10
codar.savanna.machines 10
codar.savanna.main 13
codar.savanna.model 14
codar.savanna.node_layout 16
codar.savanna.producer 16
codar.savanna.runners 16
codar.savanna.scheduler 17
codar.savanna.status 17
codar.savanna.summit_helper 20

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- Exception
  - codar.savanna.exc.SavannaException 69
    - codar.savanna.exc.MachineNotFound 31
- codar.savanna.machines.MachineNode 29
  - codar.savanna.machines.SummitNode 70
- codar.savanna.model.NodeConfig 35
- object
  - codar.savanna.consumer.PipelineRunner 49
  - codar.savanna.machines.Machine 25
  - codar.savanna.model.Pipeline 41
  - codar.savanna.node_layout.NodeLayout 36
  - codar.savanna.producer.JSONFilePipelineReader 23
  - codar.savanna.runners.Runner 68
    - codar.savanna.runners.MPIRunner 32
    - codar.savanna.runners.SummitRunner 72
  - codar.savanna.scheduler.JobList 21
  - codar.savanna.status.PipelineState 55
- Thread
  - codar.savanna.model.Run 57
  - codar.savanna.status.WorkflowStatus 76

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

codar.savanna.scheduler.JobList 21
codar.savanna.producer.JSONFilePipelineReader 23
codar.savanna.machines.Machine 25
codar.savanna.machines.MachineNode 29
codar.savanna.exc.MachineNotFound 31
codar.savanna.runners.MPIRunner 32
codar.savanna.model.NodeConfig 35
codar.savanna.node_layout.NodeLayout 36
codar.savanna.model.Pipeline 41
codar.savanna.consumer.PipelineRunner 49
codar.savanna.status.PipelineState 55
codar.savanna.model.Run 57
codar.savanna.runners.Runner 68
codar.savanna.exc.SavannaException 69
codar.savanna.machines.SummitNode 70
codar.savanna.runners.SummitRunner 72
codar.savanna.status.WorkflowStatus 76

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

__init__.py 79
consumer.py 79
exc.py 79
machines.py 80
main.py 80
model.py 80
node_layout.py 81
producer.py 81
runners.py 81
scheduler.py 82
status.py 82
summit_helper.py 82

Chapter 5

Namespace Documentation

5.1 codar Namespace Reference

Namespaces

- savanna

5.2 codar.savanna Namespace Reference

Namespaces

- consumer
- exc
- machines
- main
- model
- node_layout
- producer
- runners
- scheduler
- status
- summit_helper

5.2.1 Detailed Description

```
Classes for running pipelines of MPI tasks based on a specified
total process limit. The system is designed to use two + N threads:

1. consumer thread: get pipelines from queue and execute them when process
   slots become available. Stops when a None pipeline is received.

2. producer thread: add pipelines to queue. Can be from file or from network
   service.

3. monitor threads: each process spawned by the consumer thread has a monitor
   thread that blocks on the processes completing with a timeout, and kills the
   process if it’s not done after the timeout is reached.
```

5.3 codar.savanna.consumer Namespace Reference

Classes

- class PipelineRunner

5.3.1 Detailed Description

```
Classes for 'consuming' pipelines - running groups of MPI tasks based on a
specified total process limit.
```

5.4 codar.savanna.exc Namespace Reference

Classes

- class MachineNotFound
- class SavannaException

5.4.1 Detailed Description

```
Exceptions.
```

5.5 codar.savanna.machines Namespace Reference

Classes

- class Machine
- class MachineNode
- class SummitNode

Functions

- def get_by_name (name)

Variables

- SCHEDULER_OPTIONS = set(["project", "queue", "constraint", "license"])
- local = Machine('local', "local", "mpiexec", MachineNode, processes_per_node=1)
- titan
- cori
- theta
- summit

5.5.1 Detailed Description

```
Configuration for machines supported by Codar.
```

5.5.2 Function Documentation

5.5.2.1 get_by_name()

```
def codar.savanna.machines.get_by_name (
              name )
```

Definition at line 149 of file machines.py.

5.5.3 Variable Documentation

5.5.3.1 cori

```
codar.savanna.machines.cori
```

Initial value:

```
=  Machine('cori', "slurm", "srun", MachineNode,
                processes_per_node=32, node_exclusive=True,
                dataspaces_servers_per_node=4,
                scheduler_options=dict(project="",
                                       queue="debug",
                                       constraint="haswell",
                                       license="SCRATCH,project"))
```

Definition at line 128 of file machines.py.

5.5.3.2 local

```
codar.savanna.machines.local = Machine('local', "local", "mpiexec", MachineNode, processes_←
per_node=1)
```

Definition at line 118 of file machines.py.

5.5.3.3 SCHEDULER_OPTIONS

```
codar.savanna.machines.SCHEDULER_OPTIONS = set(["project", "queue", "constraint", "license"])
```

Definition at line 13 of file machines.py.

5.5.3.4 summit

```
codar.savanna.machines.summit
```

Initial value:

```
=  Machine('summit', "ibm_lsf", "jsrun", SummitNode,
                  processes_per_node=42, node_exclusive=True,
                  scheduler_options=dict(project=""))
```

Definition at line 144 of file machines.py.

5.5.3.5 theta

```
codar.savanna.machines.theta
```

Initial value:

```
=  Machine('theta', "cobalt", "aprun", MachineNode,
                 processes_per_node=64, node_exclusive=True,
                 dataspaces_servers_per_node=8,
                 scheduler_options=dict(project="",
                                        queue="debug-flat-quad"))
```

Definition at line 137 of file machines.py.

5.5.3.6 titan

```
codar.savanna.machines.titan
```

Initial value:

```
=  Machine('titan', "pbs", "aprun", MachineNode,
                 processes_per_node=16, node_exclusive=True,
                 scheduler_options=dict(project="", queue="debug"),
                 dataspaces_servers_per_node=4)
```

Definition at line 120 of file machines.py.

5.6 codar.savanna.main Namespace Reference

Functions

- def parse_args ()
- def main ()
- def get_job_id ()

Variables

- consumer = None

5.6.1 Detailed Description

```
Main program for executing workflow script with different producers and
runners.
```

5.6.2 Function Documentation

5.6.2.1 get_job_id()

```
def codar.savanna.main.get_job_id ( )
```

Definition at line 104 of file main.py.

5.6.2.2 main()

```
def codar.savanna.main.main ( )
```

Definition at line 39 of file main.py.

5.6.2.3 parse_args()

```
def codar.savanna.main.parse_args ( )
```

Definition at line 18 of file main.py.

5.6.3 Variable Documentation

5.6.3.1 consumer

```
codar.savanna.main.consumer = None
```

Definition at line 15 of file main.py.

5.7 codar.savanna.model Namespace Reference

Classes

- class NodeConfig
- class Pipeline
- class Run

Variables

- string STDOUT_NAME = 'codar.workflow.stdout'
- string STDERR_NAME = 'codar.workflow.stderr'
- string RETURN_NAME = 'codar.workflow.return'
- string WALLTIME_NAME = 'codar.workflow.walltime'
- int KILL_WAIT = 30
- int WAIT_DELAY_KILL = 30
- int WAIT_DELAY_GIVE_UP = 120

5.7.1 Detailed Description

```
Classes for tracking pipelines and the runs within each pipeline in separate
monitor threads that synchronize state.

Note that there is state tracked in these classes which is not available just
by looking at the return code. In particular, a run my be killed for several
different reasons: external signal, run timeout reached, other run in pipeline
failed (when kill on partial fail is set), or if the entire workflow is killed.

The goal here is to provide as much information as possible about why a
pipeline failed, to make an informed decision about whether it is worth
running again when the workflow is restarted, or if it's failure was more
permanent and not subject to outside forces like the job walltime expiring.
```

5.7.2 Variable Documentation

5.7.2.1 KILL_WAIT

```
int codar.savanna.model.KILL_WAIT = 30
```

Definition at line 37 of file model.py.

5.7.2.2 RETURN_NAME

```
string codar.savanna.model.RETURN_NAME = 'codar.workflow.return'
```

Definition at line 34 of file model.py.

5.7.2.3 STDERR_NAME

```
string codar.savanna.model.STDERR_NAME = 'codar.workflow.stderr'
```

Definition at line 33 of file model.py.

5.7.2.4 STDOUT_NAME

```
string codar.savanna.model.STDOUT_NAME = 'codar.workflow.stdout'
```

Definition at line 32 of file model.py.

5.7.2.5 WAIT_DELAY_GIVE_UP

```
int codar.savanna.model.WAIT_DELAY_GIVE_UP = 120
```

Definition at line 39 of file model.py.

5.7.2.6 WAIT_DELAY_KILL

```
int codar.savanna.model.WAIT_DELAY_KILL = 30
```

Definition at line 38 of file model.py.

5.7.2.7 WALLTIME_NAME

```
string codar.savanna.model.WALLTIME_NAME = 'codar.workflow.walltime'
```

Definition at line 35 of file model.py.

5.8 codar.savanna.node_layout Namespace Reference

Classes

- class NodeLayout

5.9 codar.savanna.producer Namespace Reference

Classes

- class JSONFilePipelineReader

5.9.1 Detailed Description

```
Classes for producing pipelines.
```

5.10 codar.savanna.runners Namespace Reference

Classes

- class MPIRunner
- class Runner
- class SummitRunner

Variables

- mpiexec = MPIRunner('mpiexec', '-n', hostfile='--hostfile')
- aprun = MPIRunner('aprun', '-n', tasks_per_node_arg='-N', hostfile='-L')
- srun = MPIRunner('srun', '-n', nodes_arg='-N', hostfile='-w')
- jsrun = SummitRunner()

5.10.1 Variable Documentation

5.10.1.1 aprun

```
codar.savanna.runners.aprun = MPIRunner('aprun', '-n', tasks_per_node_arg='-N', hostfile='-L')
```

Definition at line 94 of file runners.py.

5.10.1.2 jsrun

```
codar.savanna.runners.jsrun = SummitRunner()
```

Definition at line 96 of file runners.py.

5.10.1.3 mpiexec

```
codar.savanna.runners.mpiexec = MPIRunner('mpiexec', '-n', hostfile='--hostfile')
```

Definition at line 93 of file runners.py.

5.10.1.4 srun

```
codar.savanna.runners.srun = MPIRunner('srun', '-n', nodes_arg='-N', hostfile='-w')
```

Definition at line 95 of file runners.py.

5.11 codar.savanna.scheduler Namespace Reference

Classes

- class JobList

5.11.1 Detailed Description

```
Classes related to finding a job that can run on available resources. Does
not assume any knowledge of how long each job will take. Designed for greedy
search of a job that will fit whenever resources are freed.

In the context of Cheetah workflows, it’s unlikely that there will be more than
a few hundred jobs, so it’s not worth optimizing the python search code very
much. It is however worth making sure that a job is run when resources are
available, since super computer resources are expensive. Basically it’s worth
doing some work in python to make sure we start a big unit of work on compute
nodes.
```

5.12 codar.savanna.status Namespace Reference

Classes

- class PipelineState
- class WorkflowStatus

Variables

- string NOT_STARTED = 'not_started'
- string RUNNING = 'running'
- string DONE = 'done'
- string KILLED = 'killed'
- string REASON_TIMEOUT = 'timeout'
- string REASON_FAILED = 'failed'
- string REASON_SUCCEEDED = 'succeeded'
- string REASON_EXCEPTION = 'exception'
- string REASON_NOFIT = 'nofit'

5.12.1 Detailed Description

```
Class for maintaining state of all FOB runs that the workflow consumer is
managing. State is saved in a JSON file, overwritten on each state change.
```

5.12.2 Variable Documentation

5.12.2.1 DONE

```
string codar.savanna.status.DONE = 'done'
```

Definition at line 14 of file status.py.

5.12.2.2 KILLED

```
string codar.savanna.status.KILLED = 'killed'
```

Definition at line 15 of file status.py.

5.12.2.3 NOT_STARTED

```
string codar.savanna.status.NOT_STARTED = 'not_started'
```

Definition at line 12 of file status.py.

5.12.2.4 REASON_EXCEPTION

```
string codar.savanna.status.REASON_EXCEPTION = 'exception'
```

Definition at line 20 of file status.py.

5.12.2.5 REASON_FAILED

```
string codar.savanna.status.REASON_FAILED = 'failed'
```

Definition at line 18 of file status.py.

5.12.2.6 REASON_NOFIT

```
string codar.savanna.status.REASON_NOFIT = 'nofit'
```

Definition at line 21 of file status.py.

5.12.2.7 REASON_SUCCEEDED

```
string codar.savanna.status.REASON_SUCCEEDED = 'succeeded'
```

Definition at line 19 of file status.py.

5.12.2.8 REASON_TIMEOUT

```
string codar.savanna.status.REASON_TIMEOUT = 'timeout'
```

Definition at line 17 of file status.py.

5.12.2.9 RUNNING

```
string codar.savanna.status.RUNNING = 'running'
```

Definition at line 13 of file status.py.

5.13 codar.savanna.summit_helper Namespace Reference

Functions

- def get_nodes_reqd (res_set, nrs)
- def create_erf_file (run)

5.13.1 Function Documentation

5.13.1.1 create_erf_file()

```
def codar.savanna.summit_helper.create_erf_file (
              run )
```

Definition at line 12 of file summit_helper.py.

5.13.1.2 get_nodes_reqd()

```
def codar.savanna.summit_helper.get_nodes_reqd (
              res_set,
              nrs )
```

```
Get the no. of nodes that will be required based on the resource set
and the no. of resource sets
```

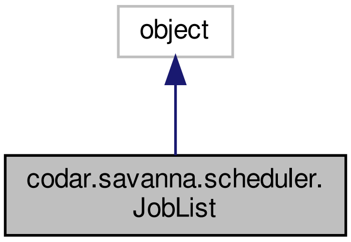
Definition at line 5 of file summit_helper.py.

Chapter 6

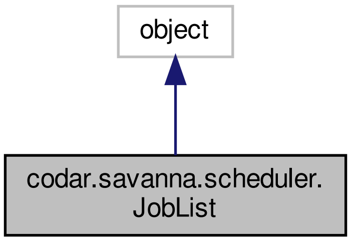
Class Documentation

6.1 codar.savanna.scheduler.JobList Class Reference

Inheritance diagram for codar.savanna.scheduler.JobList:



Collaboration diagram for codar.savanna.scheduler.JobList:



Public Member Functions

- def __init__ (self, costfn, initial_jobs=None)
- def add_job (self, job)
- def pop_job (self, max_cost)
- def __len__ (self)

6.1.1 Detailed Description

```
Manage a job list that can find and remove the highest cost job that
doesn't exceed max_cost and insert new jobs.

The job objects can be any type, but a key function must be provided
that takes an instance of a job and returns it's cost.

Uses a coordinated pair of sort list for costs and jobs, along with
the bisect module. A linked list might be more efficient, since the
list copy on insert and delete may dominate the time to do a linear
search of a small list, but it's likely fine either way for the
sizes we will encounter.
```

Definition at line 18 of file scheduler.py.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def codar.savanna.scheduler.JobList.__init__ (
              self,
              costfn,
              initial_jobs = None )
```

Definition at line 30 of file scheduler.py.

6.1.3 Member Function Documentation

6.1.3.1 __len__()

```
def codar.savanna.scheduler.JobList.__len__ (
              self )
```

Definition at line 63 of file scheduler.py.

6.1.3.2 add_job()

```
def codar.savanna.scheduler.JobList.add_job (
              self,
              job )
```

Definition at line 41 of file scheduler.py.

6.1.3.3 pop_job()

```
def codar.savanna.scheduler.JobList.pop_job (
              self,
              max_cost )
```

```
Get the highest cost job that doesn't exceed max_cost, and remove
it from the job list. Raises IndexError if the job list is empty,
returns None if no suitable jobs exist in the job list.
```

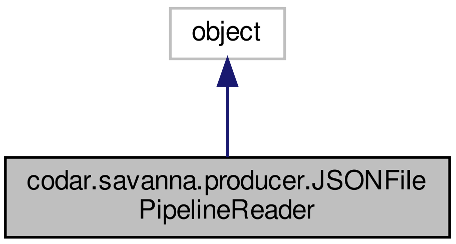
Definition at line 48 of file scheduler.py.

The documentation for this class was generated from the following file:

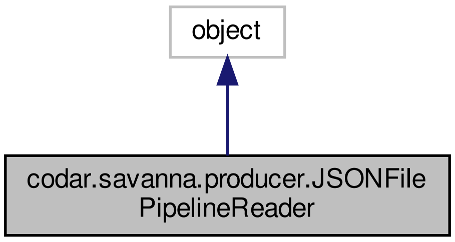
- scheduler.py

6.2 codar.savanna.producer.JSONFilePipelineReader Class Reference

Inheritance diagram for codar.savanna.producer.JSONFilePipelineReader:



Collaboration diagram for codar.savanna.producer.JSONFilePipelineReader:



Public Member Functions

- def __init__ (self, file_path)
- def read_pipelines (self)

Public Attributes

- file_path

6.2.1 Detailed Description

```
Load pipelines from a file formatted as a new line separated list of
JSON documents. Each JSON document must be a list containing dictionaries,
each dictionary discribing a code to run as part of the pipeline.
```

Definition at line 12 of file producer.py.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 __init__()

```
def codar.savanna.producer.JSONFilePipelineReader.__init__ (
              self,
              file_path )
```

Definition at line 17 of file producer.py.

6.2.3 Member Function Documentation

6.2.3.1 read_pipelines()

```
def codar.savanna.producer.JSONFilePipelineReader.read_pipelines (
              self )
```

Definition at line 20 of file producer.py.

6.2.4 Member Data Documentation

6.2.4.1 file_path

```
codar.savanna.producer.JSONFilePipelineReader.file_path
```

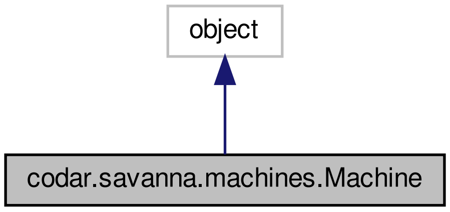
Definition at line 18 of file producer.py.

The documentation for this class was generated from the following file:

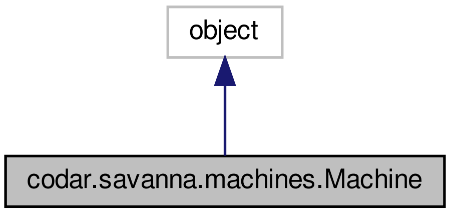
- producer.py

6.3 codar.savanna.machines.Machine Class Reference

Inheritance diagram for codar.savanna.machines.Machine:



Collaboration diagram for codar.savanna.machines.Machine:



Public Member Functions

- def __init__ (self, name, scheduler_name, runner_name, node_class, processes_per_node=None, node_exclusive=False, scheduler_options=None, dataspaces_servers_per_node=1)
- def get_scheduler_options (self, options)
- def get_nodes_reqd (self)

Public Attributes

- name
- scheduler_name
- runner_name
- node_class
- processes_per_node
- node_exclusive
- scheduler_options
- dataspaces_servers_per_node

6.3.1 Detailed Description

```
Class to represent configuration of a specific Supercomputer or
workstation, including the scheduler and runner used by the machine.
This can be used to map an experiment to run on the machine without
having to define machine specific parameter for every experiment
separately.
```

Definition at line 69 of file machines.py.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 __init__()

```
def codar.savanna.machines.Machine.__init__ (
              self,
              name,
              scheduler_name,
              runner_name,
              node_class,
              processes_per_node = None,
              node_exclusive = False,
              scheduler_options = None,
              dataspaces_servers_per_node = 1 )
```

Definition at line 78 of file machines.py.

6.3.3 Member Function Documentation

6.3.3.1 get_nodes_reqd()

```
def codar.savanna.machines.Machine.get_nodes_reqd (
              self )
```

Definition at line 100 of file machines.py.

6.3.3.2 get_scheduler_options()

```
def codar.savanna.machines.Machine.get_scheduler_options (
              self,
              options )
```

```
Validate supplied options and add default values where missing.
Returns a new dictionary.
```

Definition at line 91 of file machines.py.

6.3.4 Member Data Documentation

6.3.4.1 dataspaces_servers_per_node

```
codar.savanna.machines.Machine.dataspaces_servers_per_node
```

Definition at line 89 of file machines.py.

6.3.4.2 name

```
codar.savanna.machines.Machine.name
```

Definition at line 79 of file machines.py.

6.3.4.3 node_class

```
codar.savanna.machines.Machine.node_class
```

Definition at line 82 of file machines.py.

6.3.4.4 node_exclusive

```
codar.savanna.machines.Machine.node_exclusive
```

Definition at line 86 of file machines.py.

6.3.4.5 processes_per_node

```
codar.savanna.machines.Machine.processes_per_node
```

Definition at line 85 of file machines.py.

6.3.4.6 runner_name

```
codar.savanna.machines.Machine.runner_name
```

Definition at line 81 of file machines.py.

6.3.4.7 scheduler_name

```
codar.savanna.machines.Machine.scheduler_name
```

Definition at line 80 of file machines.py.

6.3.4.8 scheduler_options

```
codar.savanna.machines.Machine.scheduler_options
```

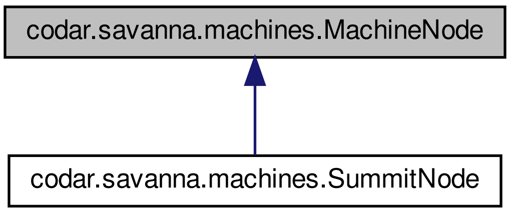
Definition at line 88 of file machines.py.

The documentation for this class was generated from the following file:

- machines.py

6.4 codar.savanna.machines.MachineNode Class Reference

Inheritance diagram for codar.savanna.machines.MachineNode:


codar.savanna.machines.MachineNode
codar.savanna.machines.SummitNode

Public Member Functions

- def __init__ (self, num_cpus, num_gpus)
- def validate_layout (self)
- def to_json (self)

Public Attributes

- cpu
- gpu

6.4.1 Detailed Description

Definition at line 16 of file machines.py.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 __init__()

```
def codar.savanna.machines.MachineNode.__init__ (
              self,
              num_cpus,
              num_gpus )
```

Definition at line 17 of file machines.py.

6.4.3 Member Function Documentation

6.4.3.1 to_json()

```
def codar.savanna.machines.MachineNode.to_json (
              self )
```

Definition at line 25 of file machines.py.

6.4.3.2 validate_layout()

```
def codar.savanna.machines.MachineNode.validate_layout (
              self )
```

Definition at line 22 of file machines.py.

6.4.4 Member Data Documentation

6.4.4.1 cpu

```
codar.savanna.machines.MachineNode.cpu
```

Definition at line 19 of file machines.py.

6.4.4.2 gpu

```
codar.savanna.machines.MachineNode.gpu
```

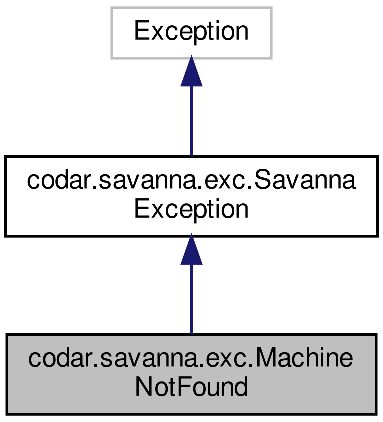
Definition at line 20 of file machines.py.

The documentation for this class was generated from the following file:

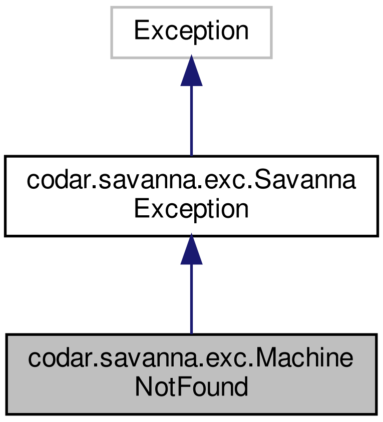
- machines.py

6.5 codar.savanna.exc.MachineNotFound Class Reference

Inheritance diagram for codar.savanna.exc.MachineNotFound:


Exception
codar.savanna.exc.Savanna Exception
codar.savanna.exc.Machine NotFound

Collaboration diagram for codar.savanna.exc.MachineNotFound:


Exception
codar.savanna.exc.Savanna Exception
codar.savanna.exc.Machine NotFound

Public Member Functions

- def __init__ (self, machine_name)

6.5.1 Detailed Description

Definition at line 10 of file exc.py.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 __init__()

```
def codar.savanna.exc.MachineNotFound.__init__ (
              self,
              machine_name )
```

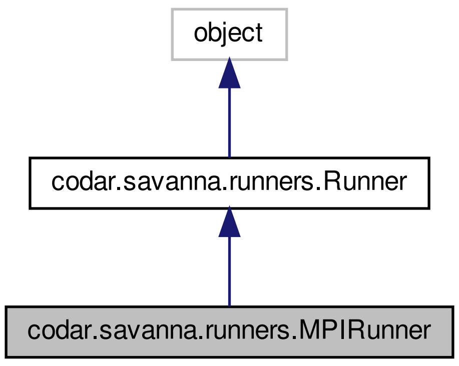
Definition at line 11 of file exc.py.

The documentation for this class was generated from the following file:

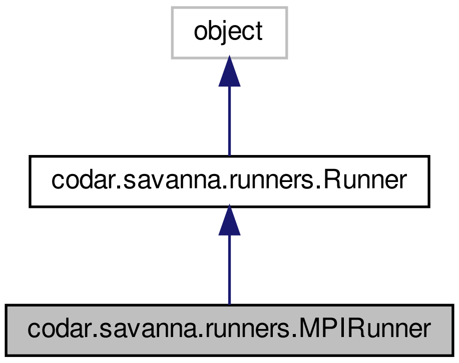
- exc.py

6.6 codar.savanna.runners.MPIRunner Class Reference

Inheritance diagram for codar.savanna.runners.MPIRunner:



Collaboration diagram for codar.savanna.runners.MPIRunner:



Public Member Functions

- def __init__ (self, exe, nprocs_arg, nodes_arg=None, tasks_per_node_arg=None, hostfile=None)
- def wrap (self, run, sched_args, find_in_path=True)

Public Attributes

- exe
- nprocs_arg
- nodes_arg
- tasks_per_node_arg
- hostfile

6.6.1 Detailed Description

Definition at line 11 of file runners.py.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 __init__()

```
def codar.savanna.runners.MPIRunner.__init__ (
              self,
              exe,
              nprocs_arg,
              nodes_arg = None,
              tasks_per_node_arg = None,
              hostfile = None )
```

Definition at line 13 of file runners.py.

6.6.3 Member Function Documentation

6.6.3.1 wrap()

```
def codar.savanna.runners.MPIRunner.wrap (
              self,
              run,
              sched_args,
              find_in_path = True )
```

Definition at line 20 of file runners.py.

6.6.4 Member Data Documentation

6.6.4.1 exe

```
codar.savanna.runners.MPIRunner.exe
```

Definition at line 14 of file runners.py.

6.6.4.2 hostfile

```
codar.savanna.runners.MPIRunner.hostfile
```

Definition at line 18 of file runners.py.

6.6.4.3 nodes_arg

```
codar.savanna.runners.MPIRunner.nodes_arg
```

Definition at line 16 of file runners.py.

6.6.4.4 nprocs_arg

```
codar.savanna.runners.MPIRunner.nprocs_arg
```

Definition at line 15 of file runners.py.

6.6.4.5 tasks_per_node_arg

```
codar.savanna.runners.MPIRunner.tasks_per_node_arg
```

Definition at line 17 of file runners.py.

The documentation for this class was generated from the following file:

- runners.py

6.7 codar.savanna.model.NodeConfig Class Reference

Public Member Functions

- def __init__ (self)

Public Attributes

- num_ranks_per_node
- cpu
- gpu

6.7.1 Detailed Description

Definition at line 52 of file model.py.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 __init__()

```
def codar.savanna.model.NodeConfig.__init__ (
              self )
```

```
Intended to look like
cpu = [ 0=[], 1=[], 2=[], 3=[] ]
gpu = [ 0=[], 1=[], 2=[], 3=[] ]
```

Definition at line 53 of file model.py.

6.7.3 Member Data Documentation

6.7.3.1 cpu

```
codar.savanna.model.NodeConfig.cpu
```

Definition at line 60 of file model.py.

6.7.3.2 gpu

```
codar.savanna.model.NodeConfig.gpu
```

Definition at line 61 of file model.py.

6.7.3.3 num_ranks_per_node

```
codar.savanna.model.NodeConfig.num_ranks_per_node
```

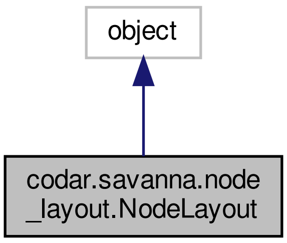
Definition at line 59 of file model.py.

The documentation for this class was generated from the following file:

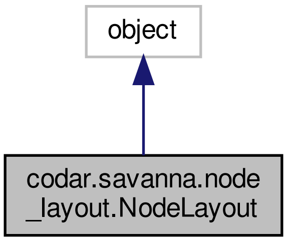
- model.py

6.8 codar.savanna.node_layout.NodeLayout Class Reference

Inheritance diagram for codar.savanna.node_layout.NodeLayout:



Collaboration diagram for codar.savanna.node_layout.NodeLayout:



Public Member Functions

- def __init__ (self, layout_list)
- def add_node (self, node_dict)
- def get_node_containing_code (self, code)
- def codes_per_node (self)
- def shared_nodes (self)
- def ppn (self)
- def validate (self, ppn, codes_per_node, shared_nodes)
- def as_data_list (self)
- def serialize_to_dict (self)
- def copy (self)
- def group_codes_by_node (self)
- def populate_remaining (self, rc_names, ppn)
- def default_no_share_layout (cls, ppn, code_names)

Public Attributes

- layout_list
- layout_map

6.8.1 Detailed Description

```
Class representing options on how to organize a multi-exe task across
many nodes. It is the scheduler model's job to take this and produce the
correct scheduler and runner options to make this happen, or raise an error
if it's not possible. Note that this will generally be different for each
machine unless it is very simple and suppored uniformly by all desired
machines.

A layout is represented as a list of dictionaries, where each dictionary
described codes to be run together on a single node. The keys are
the names of the codes, and the values are the number of processes to
assign to each.
```

Definition at line 6 of file node_layout.py.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 __init__()

```
def codar.savanna.node_layout.NodeLayout.__init__ (
              self,
              layout_list )
```

Definition at line 20 of file node_layout.py.

6.8.3 Member Function Documentation

6.8.3.1 add_node()

```
def codar.savanna.node_layout.NodeLayout.add_node (
              self,
              node_dict )
```

```
Add a node to an existing layout, e.g. add sosflow.
```

Definition at line 43 of file node_layout.py.

6.8.3.2 as_data_list()

```
def codar.savanna.node_layout.NodeLayout.as_data_list (
              self )
```

Definition at line 114 of file node_layout.py.

6.8.3.3 codes_per_node()

```
def codar.savanna.node_layout.NodeLayout.codes_per_node (
              self )
```

Definition at line 55 of file node_layout.py.

6.8.3.4 copy()

```
def codar.savanna.node_layout.NodeLayout.copy (
              self )
```

Definition at line 129 of file node_layout.py.

6.8.3.5 default_no_share_layout()

```
def codar.savanna.node_layout.NodeLayout.default_no_share_layout (
              cls,
              ppn,
              code_names )
```

```
Create a layout object for the specified codes and ppn, where each
code uses max procs on it's own node.
```

Definition at line 173 of file node_layout.py.

6.8.3.6 get_node_containing_code()

```
def codar.savanna.node_layout.NodeLayout.get_node_containing_code (
              self,
              code )
```

```
Get node dict containing the specified code. Raises KeyError if
not found.
```

Definition at line 50 of file node_layout.py.

6.8.3.7 group_codes_by_node()

```
def codar.savanna.node_layout.NodeLayout.group_codes_by_node (
              self )
```

```
Return a list of dicts, where each list represents codes on a
node, and a dict key for ppn
Example: [ {sim,analysis1}, {analysis2}, {viz} ].
Must take Summit NodeConfigs into account
```

Definition at line 132 of file node_layout.py.

6.8.3.8 populate_remaining()

```
def codar.savanna.node_layout.NodeLayout.populate_remaining (
              self,
              rc_names,
              ppn )
```

Definition at line 161 of file node_layout.py.

6.8.3.9 ppn()

```
def codar.savanna.node_layout.NodeLayout.ppn (
              self )
```

Definition at line 61 of file node_layout.py.

6.8.3.10 serialize_to_dict()

```
def codar.savanna.node_layout.NodeLayout.serialize_to_dict (
              self )
```

```
Get a copy of the data list passed to the constructor,
suitable for JSON serialization.
```

Definition at line 117 of file node_layout.py.

6.8.3.11 shared_nodes()

```
def codar.savanna.node_layout.NodeLayout.shared_nodes (
              self )
```

Definition at line 58 of file node_layout.py.

6.8.3.12 validate()

```
def codar.savanna.node_layout.NodeLayout.validate (
              self,
              ppn,
              codes_per_node,
              shared_nodes )
```

```
Given a machine ppn and max numer of codes (e.g. 4 on cori),
raise a ValueError if the specified layout won’t fit.
Dont modify this yet, this is being used by the tests
```

Definition at line 96 of file node_layout.py.

6.8.4 Member Data Documentation

6.8.4.1 layout_list

`codar.savanna.node_layout.NodeLayout.layout_list`

Definition at line 34 of file node_layout.py.

6.8.4.2 layout_map

`codar.savanna.node_layout.NodeLayout.layout_map`

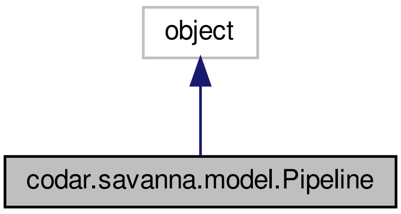
Definition at line 35 of file node_layout.py.

The documentation for this class was generated from the following file:

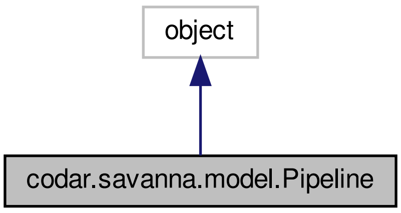
- node_layout.py

6.9 codar.savanna.model.Pipeline Class Reference

Inheritance diagram for codar.savanna.model.Pipeline:



Collaboration diagram for codar.savanna.model.Pipeline:



Public Member Functions

- def __init__ (self, pipe_id, runs, working_dir, total_nodes, machine_name, kill_on_partial_failure=False, post_process_script=None, post_process_args=None, post_process_stop_on_failure=False, node_↵ layout=None, launch_mode=None)
- def from_data (cls, data)
- def start (self, consumer, nodes_assigned, runner=None)
- def run_finished (self, run)
- def run_post_process_script (self)
- def add_done_callback (self, fn)
- def remove_done_callback (self, fn)
- def add_fatal_callback (self, fn)
- def remove_fatal_callback (self, fn)
- def get_nodes_used (self)
- def set_ppn (self, ppn)
- def set_total_nodes (self)
- def get_state (self)
- def get_pids (self)
- def force_kill_all (self)
- def join_all (self)

Public Attributes

- id
- runs
- working_dir
- kill_on_partial_failure
- post_process_script
- post_process_args
- post_process_stop_on_failure
- node_layout
- machine_name
- done_callbacks
- fatal_callbacks
- total_procs
- log_prefix
- total_nodes
- launch_mode
- nodes_assigned

6.9.1 Detailed Description

Definition at line 449 of file model.py.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 __init__()

```
def codar.savanna.model.Pipeline.__init__ (
              self,
              pipe_id,
              runs,
              working_dir,
              total_nodes,
              machine_name,
              kill_on_partial_failure = False,
              post_process_script = None,
              post_process_args = None,
              post_process_stop_on_failure = False,
              node_layout = None,
              launch_mode = None )
```

Definition at line 455 of file model.py.

6.9.3 Member Function Documentation

6.9.3.1 add_done_callback()

```
def codar.savanna.model.Pipeline.add_done_callback (
              self,
              fn )
```

Definition at line 818 of file model.py.

6.9.3.2 add_fatal_callback()

```
def codar.savanna.model.Pipeline.add_fatal_callback (
              self,
              fn )
```

Definition at line 830 of file model.py.

6.9.3.3 force_kill_all()

```
def codar.savanna.model.Pipeline.force_kill_all (
              self )
```

```
Kill all runs and don't run post processing. Note that this call may
block waiting for all runs to be started, to avoid confusing races.
If the pipeline is already done, this does nothing. If one or more
runs are still active, or have not yet been marked as finished, then
it will mark the entire pipeline as killed so it can be re-run from
scratch on a restart if desired.
```

Definition at line 912 of file model.py.

6.9.3.4 from_data()

```
def codar.savanna.model.Pipeline.from_data (
              cls,
              data )
```

```
Create Pipeline instance from dictionary data structure, containing
at least "id" and "runs" keys. The "runs" key must have a list of dict,
and each dict is parsed using Run.from_data.
Raises KeyError if a required key is missing.
```

Definition at line 497 of file model.py.

6.9.3.5 get_nodes_used()

```
def codar.savanna.model.Pipeline.get_nodes_used (
              self )
```

Definition at line 850 of file model.py.

6.9.3.6 get_pids()

```
def codar.savanna.model.Pipeline.get_pids (
              self )
```

Definition at line 908 of file model.py.

6.9.3.7 get_state()

```
def codar.savanna.model.Pipeline.get_state (
              self )
```

Definition at line 883 of file model.py.

6.9.3.8 join_all()

```
def codar.savanna.model.Pipeline.join_all (
              self )
```

Definition at line 933 of file model.py.

6.9.3.9 remove_done_callback()

```
def codar.savanna.model.Pipeline.remove_done_callback (
              self,
              fn )
```

Definition at line 821 of file model.py.

6.9.3.10 remove_fatal_callback()

```
def codar.savanna.model.Pipeline.remove_fatal_callback (
              self,
              fn )
```

Definition at line 833 of file model.py.

6.9.3.11 run_finished()

```
def codar.savanna.model.Pipeline.run_finished (
              self,
              run )
```

Definition at line 744 of file model.py.

6.9.3.12 run_post_process_script()

```
def codar.savanna.model.Pipeline.run_post_process_script (
              self )
```

Definition at line 772 of file model.py.

6.9.3.13 set_ppn()

```
def codar.savanna.model.Pipeline.set_ppn (
              self,
              ppn )
```

```
Determine number of nodes needed to run pipeline with the specified
node layout or full occupancy layout with ppn. Also updates runs
to set node and task per node counts.
TODO: This should be set by Cheetah in fobs.json
```

Definition at line 855 of file model.py.

6.9.3.14 set_total_nodes()

```
def codar.savanna.model.Pipeline.set_total_nodes (
              self )
```

```
To be deprecated
```

Definition at line 877 of file model.py.

6.9.3.15 start()

```
def codar.savanna.model.Pipeline.start (
              self,
              consumer,
              nodes_assigned,
              runner = None )
```

Definition at line 550 of file model.py.

6.9.4 Member Data Documentation

6.9.4.1 done_callbacks

```
codar.savanna.model.Pipeline.done_callbacks
```

Definition at line 473 of file model.py.

6.9.4.2 fatal_callbacks

```
codar.savanna.model.Pipeline.fatal_callbacks
```

Definition at line 474 of file model.py.

6.9.4.3 id

```
codar.savanna.model.Pipeline.id
```

Definition at line 456 of file model.py.

6.9.4.4 kill_on_partial_failure

`codar.savanna.model.Pipeline.kill_on_partial_failure`

Definition at line 459 of file model.py.

6.9.4.5 launch_mode

`codar.savanna.model.Pipeline.launch_mode`

Definition at line 483 of file model.py.

6.9.4.6 log_prefix

`codar.savanna.model.Pipeline.log_prefix`

Definition at line 476 of file model.py.

6.9.4.7 machine_name

`codar.savanna.model.Pipeline.machine_name`

Definition at line 464 of file model.py.

6.9.4.8 node_layout

`codar.savanna.model.Pipeline.node_layout`

Definition at line 463 of file model.py.

6.9.4.9 nodes_assigned

`codar.savanna.model.Pipeline.nodes_assigned`

Definition at line 487 of file model.py.

6.9.4.10 post_process_args

```
codar.savanna.model.Pipeline.post_process_args
```

Definition at line 461 of file model.py.

6.9.4.11 post_process_script

```
codar.savanna.model.Pipeline.post_process_script
```

Definition at line 460 of file model.py.

6.9.4.12 post_process_stop_on_failure

```
codar.savanna.model.Pipeline.post_process_stop_on_failure
```

Definition at line 462 of file model.py.

6.9.4.13 runs

```
codar.savanna.model.Pipeline.runs
```

Definition at line 457 of file model.py.

6.9.4.14 total_nodes

```
codar.savanna.model.Pipeline.total_nodes
```

Definition at line 482 of file model.py.

6.9.4.15 total_procs

```
codar.savanna.model.Pipeline.total_procs
```

Definition at line 475 of file model.py.

6.9.4.16 working_dir

```
codar.savanna.model.Pipeline.working_dir
```

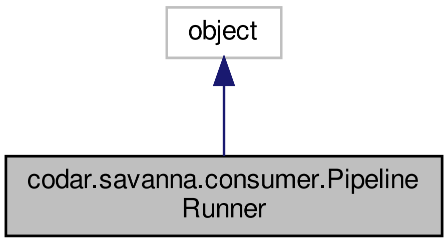
Definition at line 458 of file model.py.

The documentation for this class was generated from the following file:

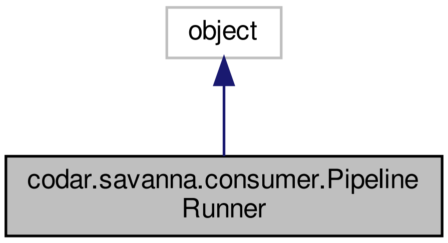
- model.py

6.10 codar.savanna.consumer.PipelineRunner Class Reference

Inheritance diagram for codar.savanna.consumer.PipelineRunner:



Collaboration diagram for codar.savanna.consumer.PipelineRunner:



Public Member Functions

- def __init__ (self, runner, max_nodes, machine_name, processes_per_node, status_file=None)
- def add_pipeline (self, p)
- def stop (self)
- def kill_all (self)
- def run_finished (self, run)
- def pipeline_finished (self, pipeline)
- def pipeline_fatal (self, pipeline)
- def run_pipelines (self)

Public Attributes

- max_nodes
- machine_name
- ppn
- runner
- job_list_cv
- job_list
- free_cv
- free_nodes
- pipelines_lock
- pipelines
- allocated_nodes

6.10.1 Detailed Description

```
Runner that assumes a homogonous set of nodes. Now only support only
node based limiting (although process limiting can be emulated by setting
process_per_node=1 and max_nodes=max_procs).

Threading model: assumes there could be multiple producer threads calling
add_pipeline, e.g. if using a dynamic job submission model based on
results of previous jobs. Pipelines and each Run in a pipeline are all
executed in separate threads, so their notification callbacks execute in
separate threads, and their threads must be joined before exiting. The
stop and kill_all methods could be called from any of the producer,
Pipeline or Run threads.
```

Definition at line 18 of file consumer.py.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 __init__()

```
def codar.savanna.consumer.PipelineRunner.__init__ (
              self,
              runner,
              max_nodes,
              machine_name,
              processes_per_node,
              status_file = None )
```

Definition at line 32 of file consumer.py.

6.10.3 Member Function Documentation

6.10.3.1 add_pipeline()

```
def codar.savanna.consumer.PipelineRunner.add_pipeline (
              self,
              p )
```

Definition at line 73 of file consumer.py.

6.10.3.2 kill_all()

```
def codar.savanna.consumer.PipelineRunner.kill_all (
              self )
```

```
Kill all running processes spawned by this consumer and don’t
start any new processes.
```

Definition at line 114 of file consumer.py.

6.10.3.3 pipeline_fatal()

```
def codar.savanna.consumer.PipelineRunner.pipeline_fatal (
              self,
              pipeline )
```

Definition at line 190 of file consumer.py.

6.10.3.4 pipeline_finished()

```
def codar.savanna.consumer.PipelineRunner.pipeline_finished (
              self,
              pipeline )
```

```
Monitor thread(s) should call this as pipelines complete.
```

Definition at line 164 of file consumer.py.

6.10.3.5 run_finished()

```
def codar.savanna.consumer.PipelineRunner.run_finished (
              self,
              run )
```

```
TO BE DEPRECATED.
Monitor thread(s) should call this as runs
complete. To be deprecated, as the functionality fails when
node_layout is set to node-sharing.

This means that for node_exclusive, resources held by a run are not
released when the run terminates. For kill_on_partial_failure=False,
this could lead to unused resources, which is ok.
```

Definition at line 148 of file consumer.py.

6.10.3.6 run_pipelines()

```
def codar.savanna.consumer.PipelineRunner.run_pipelines (
              self )
```

```
Main loop of consumer thread. Does not return until all child
threads are complete.
```

Definition at line 194 of file consumer.py.

6.10.3.7 stop()

```
def codar.savanna.consumer.PipelineRunner.stop (
              self )
```

```
Signal to stop when all pipelines are finished. Don't allow adding
new pipelines.
```

Definition at line 105 of file consumer.py.

6.10.4 Member Data Documentation

6.10.4.1 allocated_nodes

```
codar.savanna.consumer.PipelineRunner.allocated_nodes
```

Definition at line 60 of file consumer.py.

6.10.4.2 free_cv

```
codar.savanna.consumer.PipelineRunner.free_cv
```

Definition at line 47 of file consumer.py.

6.10.4.3 free_nodes

```
codar.savanna.consumer.PipelineRunner.free_nodes
```

Definition at line 48 of file consumer.py.

6.10.4.4 job_list

```
codar.savanna.consumer.PipelineRunner.job_list
```

Definition at line 45 of file consumer.py.

6.10.4.5 job_list_cv

```
codar.savanna.consumer.PipelineRunner.job_list_cv
```

Definition at line 43 of file consumer.py.

6.10.4.6 machine_name

```
codar.savanna.consumer.PipelineRunner.machine_name
```

Definition at line 34 of file consumer.py.

6.10.4.7 max_nodes

```
codar.savanna.consumer.PipelineRunner.max_nodes
```

Definition at line 33 of file consumer.py.

6.10.4.8 pipelines

```
codar.savanna.consumer.PipelineRunner.pipelines
```

Definition at line 51 of file consumer.py.

6.10.4.9 pipelines_lock

```
codar.savanna.consumer.PipelineRunner.pipelines_lock
```

Definition at line 50 of file consumer.py.

6.10.4.10 ppn

```
codar.savanna.consumer.PipelineRunner.ppn
```

Definition at line 35 of file consumer.py.

6.10.4.11 runner

```
codar.savanna.consumer.PipelineRunner.runner
```

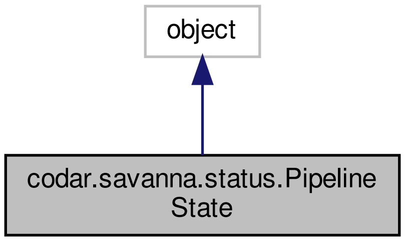
Definition at line 36 of file consumer.py.

The documentation for this class was generated from the following file:

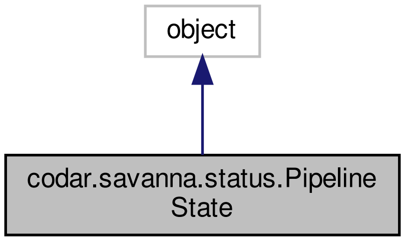
- consumer.py

6.11 codar.savanna.status.PipelineState Class Reference

Inheritance diagram for codar.savanna.status.PipelineState:



Collaboration diagram for codar.savanna.status.PipelineState:



Public Member Functions

- def __init__ (self, pipeline_id, state, reason=None, return_codes=None)
- def as_data (self)

Public Attributes

- id
- state
- reason
- return_codes

6.11.1 Detailed Description

Definition at line 48 of file status.py.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 __init__()

```
def codar.savanna.status.PipelineState.__init__ (
              self,
              pipeline_id,
              state,
              reason = None,
              return_codes = None )
```

Definition at line 49 of file status.py.

6.11.3 Member Function Documentation

6.11.3.1 as_data()

```
def codar.savanna.status.PipelineState.as_data (
              self )
```

Definition at line 55 of file status.py.

6.11.4 Member Data Documentation

6.11.4.1 id

```
codar.savanna.status.PipelineState.id
```

Definition at line 50 of file status.py.

6.11.4.2 reason

```
codar.savanna.status.PipelineState.reason
```

Definition at line 52 of file status.py.

6.11.4.3 return_codes

`codar.savanna.status.PipelineState.return_codes`

Definition at line 53 of file status.py.

6.11.4.4 state

`codar.savanna.status.PipelineState.state`

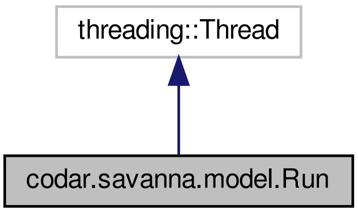
Definition at line 51 of file status.py.

The documentation for this class was generated from the following file:

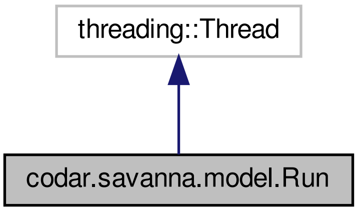
- status.py

6.12 codar.savanna.model.Run Class Reference

Inheritance diagram for codar.savanna.model.Run:



Collaboration diagram for codar.savanna.model.Run:



Public Member Functions

- def __init__ (self, name, exe, args, sched_args, env, working_dir, timeout=None, nprocs=1, res_set=None, stdout_path=None, stderr_path=None, return_path=None, walltime_path=None, log_prefix=None, sleep_↩ after=None, depends_on_runs=None, hostfile=None, runner_override=False)
- def from_data (cls, data)
- def mpmd_run (cls, runs)
- def set_runner (self, runner)
- def timed_out (self)
- def killed (self)
- def exception (self)
- def succeeded (self)
- def add_callback (self, fn)
- def remove_callback (self, fn)
- def run (self)
- def kill (self)
- def get_returncode (self)
- def get_pid (self)
- def close (self)
- def join (self)
- def get_nodes_used (self)
- def create_node_config (self)

Public Attributes

- name
- exe
- args
- sched_args
- env
- working_dir
- timeout
- nprocs
- res_set
- stdout_path
- stderr_path
- return_path
- walltime_path
- sleep_after
- log_prefix
- runner
- callbacks
- nodes
- tasks_per_node
- depends_on_runs
- hostfile
- machine
- nodes_assigned
- node_config
- erf_file
- runner_override

6.12.1 Detailed Description

```
Manage running a single executable within a pipeline. When start is
called, it will launch the process with Popen and call wait in the new
thread with a timeout, killing if the process does not finish in time.
```

Definition at line 64 of file model.py.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 __init__()

```
def codar.savanna.model.Run.__init__ (
              self,
              name,
              exe,
              args,
              sched_args,
              env,
              working_dir,
              timeout = None,
              nprocs = 1,
              res_set = None,
              stdout_path = None,
              stderr_path = None,
              return_path = None,
              walltime_path = None,
              log_prefix = None,
              sleep_after = None,
              depends_on_runs = None,
              hostfile = None,
              runner_override = False )
```

Definition at line 74 of file model.py.

6.12.3 Member Function Documentation

6.12.3.1 add_callback()

```
def codar.savanna.model.Run.add_callback (
              self,
              fn )
```

```
Function takes single argument which is this run instance, and is
called when the process is complete (either normally or killed by
timeout). Callbacks must not block.
```

Definition at line 228 of file model.py.

6.12.3.2 close()

```
def codar.savanna.model.Run.close (
              self )
```

Definition at line 426 of file model.py.

6.12.3.3 create_node_config()

```
def codar.savanna.model.Run.create_node_config (
              self )
```

Definition at line 445 of file model.py.

6.12.3.4 exception()

```
def codar.savanna.model.Run.exception (
              self )
```

```
True if there was a python exception in the run method. When this
is the case, the state of the underlying process is unknown - it may
have been started or not.
```

Definition at line 211 of file model.py.

6.12.3.5 from_data()

```
def codar.savanna.model.Run.from_data (
              cls,
              data )
```

```
Create Run instance from nested dictionary data structure, e.g.
parsed from JSON. The keys 'name', 'exe', 'args' are required, all the
other keys are optional and have the same names as the constructor
args. Raises KeyError if a required key is missing.
```

Definition at line 146 of file model.py.

6.12.3.6 get_nodes_used()

```
def codar.savanna.model.Run.get_nodes_used (
              self )
```

```
Get number of nodes needed to run this app. Requires that the
pipeline set_ppn method has been called to set this and tasks_per_node
on each run.
```

Definition at line 436 of file model.py.

6.12.3.7 get_pid()

```
def codar.savanna.model.Run.get_pid (
              self )
```

Definition at line 421 of file model.py.

6.12.3.8 get_returncode()

```
def codar.savanna.model.Run.get_returncode (
              self )
```

Definition at line 416 of file model.py.

6.12.3.9 join()

```
def codar.savanna.model.Run.join (
              self )
```

Definition at line 431 of file model.py.

6.12.3.10 kill()

```
def codar.savanna.model.Run.kill (
              self )
```

```
Kill process and cause run thread to complete after the wait
returns. If the run is already done, does nothing. If the process is
killed, it will mark the state as killed so it can be re-run on
workflow restart. Thread safe.
```

Definition at line 319 of file model.py.

6.12.3.11 killed()

```
def codar.savanna.model.Run.killed (
              self )
```

```
True if the run is done and the kill method was called. Note that
this will _NOT_ be true if an external kill signal caused the process
to exit. Raises ValueError if the run is not complete.
```

Definition at line 202 of file model.py.

6.12.3.12 mpmd_run()

```
def codar.savanna.model.Run.mpmd_run (
              cls,
              runs )
```

Definition at line 171 of file model.py.

6.12.3.13 remove_callback()

```
def codar.savanna.model.Run.remove_callback (
              self,
              fn )
```

Definition at line 234 of file model.py.

6.12.3.14 run()

```
def codar.savanna.model.Run.run (
              self )
```

Definition at line 237 of file model.py.

6.12.3.15 set_runner()

```
def codar.savanna.model.Run.set_runner (
              self,
              runner )
```

Definition at line 188 of file model.py.

6.12.3.16 succeeded()

```
def codar.savanna.model.Run.succeeded (
              self )
```

```
True if the run is done, finished normally, and had 0 return value.
Raises ValueError if the run is not complete.
```

Definition at line 218 of file model.py.

6.12.3.17 timed_out()

```
def codar.savanna.model.Run.timed_out (
              self )
```

```
True if the run is done and was killed because it exceeded the
specified run timeout. Raises ValueError if the run is not complete.
```

Definition at line 194 of file model.py.

6.12.4 Member Data Documentation

6.12.4.1 args

```
codar.savanna.model.Run.args
```

Definition at line 78 of file model.py.

6.12.4.2 callbacks

```
codar.savanna.model.Run.callbacks
```

Definition at line 116 of file model.py.

6.12.4.3 depends_on_runs

```
codar.savanna.model.Run.depends_on_runs
```

Definition at line 125 of file model.py.

6.12.4.4 env

```
codar.savanna.model.Run.env
```

Definition at line 80 of file model.py.

6.12.4.5 erf_file

```
codar.savanna.model.Run.erf_file
```

Definition at line 139 of file model.py.

6.12.4.6 exe

```
codar.savanna.model.Run.exe
```

Definition at line 77 of file model.py.

6.12.4.7 hostfile

```
codar.savanna.model.Run.hostfile
```

Definition at line 128 of file model.py.

6.12.4.8 log_prefix

```
codar.savanna.model.Run.log_prefix
```

Definition at line 114 of file model.py.

6.12.4.9 machine

```
codar.savanna.model.Run.machine
```

Definition at line 132 of file model.py.

6.12.4.10 name

```
codar.savanna.model.Run.name
```

Definition at line 76 of file model.py.

6.12.4.11 node_config

```
codar.savanna.model.Run.node_config
```

Definition at line 136 of file model.py.

6.12.4.12 nodes

```
codar.savanna.model.Run.nodes
```

Definition at line 121 of file model.py.

6.12.4.13 nodes_assigned

```
codar.savanna.model.Run.nodes_assigned
```

Definition at line 133 of file model.py.

6.12.4.14 nprocs

```
codar.savanna.model.Run.nprocs
```

Definition at line 83 of file model.py.

6.12.4.15 res_set

```
codar.savanna.model.Run.res_set
```

Definition at line 88 of file model.py.

6.12.4.16 return_path

```
codar.savanna.model.Run.return_path
```

Definition at line 94 of file model.py.

6.12.4.17 runner

```
codar.savanna.model.Run.runner
```

Definition at line 115 of file model.py.

6.12.4.18 runner_override

```
codar.savanna.model.Run.runner_override
```

Definition at line 143 of file model.py.

6.12.4.19 sched_args

```
codar.savanna.model.Run.sched_args
```

Definition at line 79 of file model.py.

6.12.4.20 sleep_after

```
codar.savanna.model.Run.sleep_after
```

Definition at line 98 of file model.py.

6.12.4.21 stderr_path

```
codar.savanna.model.Run.stderr_path
```

Definition at line 92 of file model.py.

6.12.4.22 stdout_path

```
codar.savanna.model.Run.stdout_path
```

Definition at line 90 of file model.py.

6.12.4.23 tasks_per_node

```
codar.savanna.model.Run.tasks_per_node
```

Definition at line 122 of file model.py.

6.12.4.24 timeout

```
codar.savanna.model.Run.timeout
```

Definition at line 82 of file model.py.

6.12.4.25 walltime_path

```
codar.savanna.model.Run.walltime_path
```

Definition at line 96 of file model.py.

6.12.4.26 working_dir

```
codar.savanna.model.Run.working_dir
```

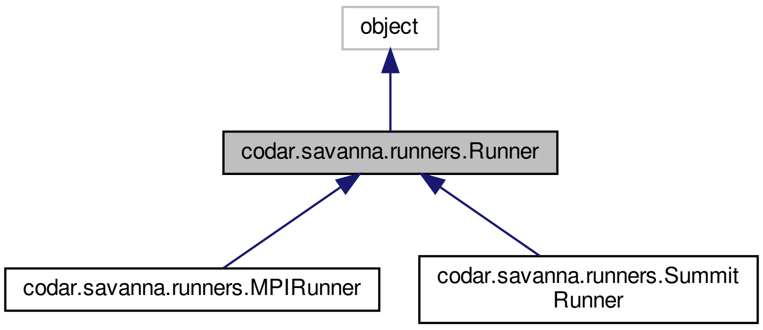
Definition at line 81 of file model.py.

The documentation for this class was generated from the following file:

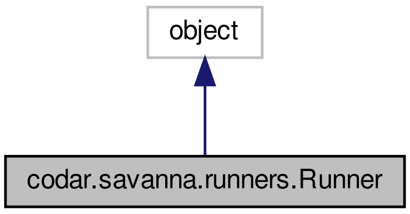
- model.py

6.13 codar.savanna.runners.Runner Class Reference

Inheritance diagram for codar.savanna.runners.Runner:


object
codar.savanna.runners.Runner
codar.savanna.runners.MPIRunner
codar.savanna.runners.Summit Runner

Collaboration diagram for codar.savanna.runners.Runner:



Public Member Functions

- def wrap (self, run, sched_args)

6.13.1 Detailed Description

Definition at line 6 of file runners.py.

6.13.2 Member Function Documentation

6.13.2.1 wrap()

```
def codar.savanna.runners.Runner.wrap (
              self,
              run,
              sched_args )
```

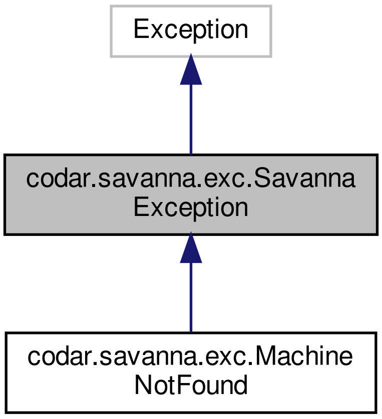
Definition at line 7 of file runners.py.

The documentation for this class was generated from the following file:

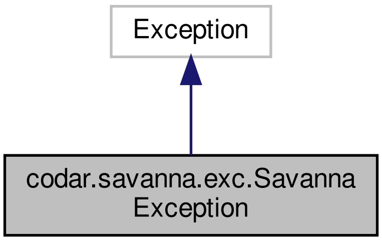
- runners.py

6.14 codar.savanna.exc.SavannaException Class Reference

Inheritance diagram for codar.savanna.exc.SavannaException:


Exception
codar.savanna.exc.Savanna Exception
codar.savanna.exc.Machine NotFound

Collaboration diagram for codar.savanna.exc.SavannaException:



6.14.1 Detailed Description

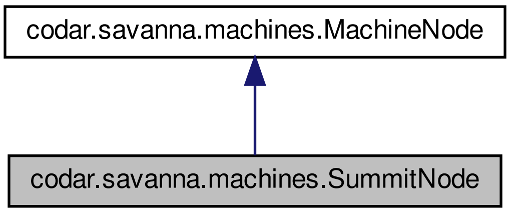
Definition at line 6 of file exc.py.

The documentation for this class was generated from the following file:

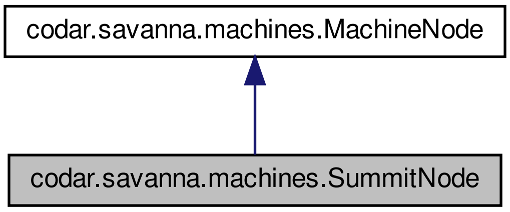
- exc.py

6.15 codar.savanna.machines.SummitNode Class Reference

Inheritance diagram for codar.savanna.machines.SummitNode:


codar.savanna.machines.MachineNode
codar.savanna.machines.SummitNode

Collaboration diagram for codar.savanna.machines.SummitNode:


codar.savanna.machines.MachineNode
codar.savanna.machines.SummitNode

Public Member Functions

- def __init__ (self)
- def validate_layout (self)
- def to_json (self)

Additional Inherited Members

6.15.1 Detailed Description

Definition at line 28 of file machines.py.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 __init__()

```
def codar.savanna.machines.SummitNode.__init__ (
              self )
```

Definition at line 29 of file machines.py.

6.15.3 Member Function Documentation

6.15.3.1 to_json()

```
def codar.savanna.machines.SummitNode.to_json (
              self )
```

Definition at line 64 of file machines.py.

6.15.3.2 validate_layout()

```
def codar.savanna.machines.SummitNode.validate_layout (
              self )
```

```
Check that 1) the same rank of the same code is not repeated,
2) a gpu is not mapped to multiple executables.
```

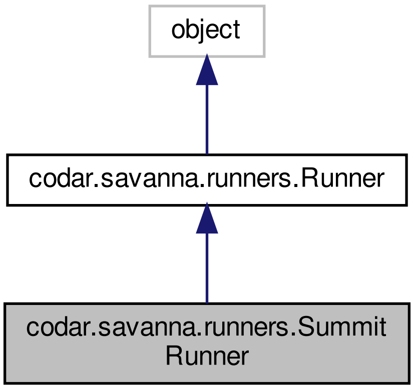
Definition at line 32 of file machines.py.

The documentation for this class was generated from the following file:

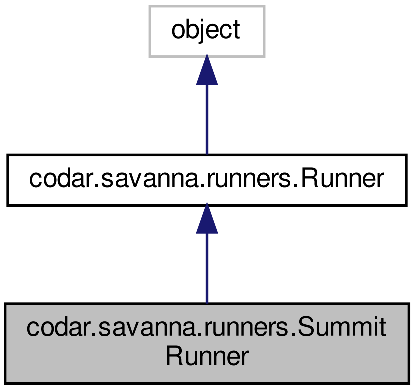
- machines.py

6.16 codar.savanna.runners.SummitRunner Class Reference

Inheritance diagram for codar.savanna.runners.SummitRunner:


object
codar.savanna.runners.Runner
codar.savanna.runners.Summit Runner

Collaboration diagram for codar.savanna.runners.SummitRunner:


object
codar.savanna.runners.Runner
codar.savanna.runners.Summit Runner

Public Member Functions

- def __init__ (self)
- def wrap (self, run, sched_args)
- def wrap_deprecated (self, run, jsrun_opts, find_in_path=True)

Public Attributes

- exe
- nrs_arg
- tasks_per_rs_arg
- cpus_per_rs_arg
- gpus_per_rs_arg
- rs_per_host_arg
- launch_distribution_arg
- bind_arg
- machine

6.16.1 Detailed Description

Definition at line 44 of file runners.py.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 __init__()

```
def codar.savanna.runners.SummitRunner.__init__ (
              self )
```

Definition at line 45 of file runners.py.

6.16.3 Member Function Documentation

6.16.3.1 wrap()

```
def codar.savanna.runners.SummitRunner.wrap (
              self,
              run,
              sched_args )
```

Definition at line 56 of file runners.py.

6.16.3.2 wrap_deprecated()

```
def codar.savanna.runners.SummitRunner.wrap_deprecated (
              self,
              run,
              jsrun_opts,
              find_in_path = True )
```

```
This function is deprecated in favor of the above that uses erf
files
```

Definition at line 60 of file runners.py.

6.16.4 Member Data Documentation

6.16.4.1 bind_arg

`codar.savanna.runners.SummitRunner.bind_arg`

Definition at line 53 of file runners.py.

6.16.4.2 cpus_per_rs_arg

`codar.savanna.runners.SummitRunner.cpus_per_rs_arg`

Definition at line 49 of file runners.py.

6.16.4.3 exe

`codar.savanna.runners.SummitRunner.exe`

Definition at line 46 of file runners.py.

6.16.4.4 gpus_per_rs_arg

`codar.savanna.runners.SummitRunner.gpus_per_rs_arg`

Definition at line 50 of file runners.py.

6.16.4.5 launch_distribution_arg

```
codar.savanna.runners.SummitRunner.launch_distribution_arg
```

Definition at line 52 of file runners.py.

6.16.4.6 machine

```
codar.savanna.runners.SummitRunner.machine
```

Definition at line 54 of file runners.py.

6.16.4.7 nrs_arg

```
codar.savanna.runners.SummitRunner.nrs_arg
```

Definition at line 47 of file runners.py.

6.16.4.8 rs_per_host_arg

```
codar.savanna.runners.SummitRunner.rs_per_host_arg
```

Definition at line 51 of file runners.py.

6.16.4.9 tasks_per_rs_arg

```
codar.savanna.runners.SummitRunner.tasks_per_rs_arg
```

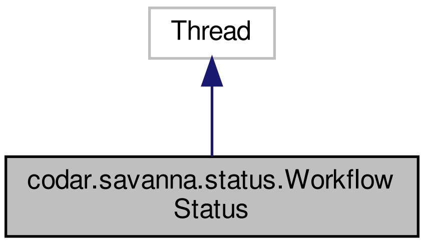
Definition at line 48 of file runners.py.

The documentation for this class was generated from the following file:

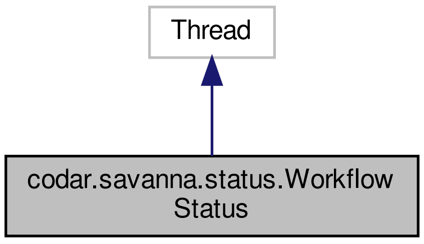
- runners.py

6.17 codar.savanna.status.WorkflowStatus Class Reference

Inheritance diagram for codar.savanna.status.WorkflowStatus:



Collaboration diagram for codar.savanna.status.WorkflowStatus:



Public Member Functions

- def __init__ (self, file_path)
- def set_state (self, pipeline_state)

Public Attributes

- file_path

6.17.1 Detailed Description

Definition at line 24 of file status.py.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 __init__()

```
def codar.savanna.status.WorkflowStatus.__init__ (
              self,
              file_path )
```

Definition at line 25 of file status.py.

6.17.3 Member Function Documentation

6.17.3.1 set_state()

```
def codar.savanna.status.WorkflowStatus.set_state (
              self,
              pipeline_state )
```

Definition at line 37 of file status.py.

6.17.4 Member Data Documentation

6.17.4.1 file_path

```
codar.savanna.status.WorkflowStatus.file_path
```

Definition at line 27 of file status.py.

The documentation for this class was generated from the following file:

- status.py

Chapter 7

File Documentation

7.1 __init__.py File Reference

Namespaces

- codar.savanna

7.2 consumer.py File Reference

Classes

- class codar.savanna.consumer.PipelineRunner

Namespaces

- codar.savanna.consumer

7.3 exc.py File Reference

Classes

- class codar.savanna.exc.SavannaException
- class codar.savanna.exc.MachineNotFound

Namespaces

- codar.savanna.exc

7.4 machines.py File Reference

Classes

- class codar.savanna.machines.MachineNode
- class codar.savanna.machines.SummitNode
- class codar.savanna.machines.Machine

Namespaces

- codar.savanna.machines

Functions

- def codar.savanna.machines.get_by_name (name)

Variables

- codar.savanna.machines.SCHEDULER_OPTIONS = set(["project", "queue", "constraint", "license"])
- codar.savanna.machines.local = Machine('local', "local", "mpiexec", MachineNode, processes_per_node=1)
- codar.savanna.machines.titan
- codar.savanna.machines.cori
- codar.savanna.machines.theta
- codar.savanna.machines.summit

7.5 main.py File Reference

Namespaces

- codar.savanna.main

Functions

- def codar.savanna.main.parse_args ()
- def codar.savanna.main.main ()
- def codar.savanna.main.get_job_id ()

Variables

- codar.savanna.main.consumer = None

7.6 model.py File Reference

Classes

- class codar.savanna.model.NodeConfig
- class codar.savanna.model.Run
- class codar.savanna.model.Pipeline

Namespaces

- codar.savanna.model

Variables

- string codar.savanna.model.STDOUT_NAME = 'codar.workflow.stdout'
- string codar.savanna.model.STDERR_NAME = 'codar.workflow.stderr'
- string codar.savanna.model.RETURN_NAME = 'codar.workflow.return'
- string codar.savanna.model.WALLTIME_NAME = 'codar.workflow.walltime'
- int codar.savanna.model.KILL_WAIT = 30
- int codar.savanna.model.WAIT_DELAY_KILL = 30
- int codar.savanna.model.WAIT_DELAY_GIVE_UP = 120

7.7 node_layout.py File Reference

Classes

- class codar.savanna.node_layout.NodeLayout

Namespaces

- codar.savanna.node_layout

7.8 producer.py File Reference

Classes

- class codar.savanna.producer.JSONFilePipelineReader

Namespaces

- codar.savanna.producer

7.9 runners.py File Reference

Classes

- class codar.savanna.runners.Runner
- class codar.savanna.runners.MPIRunner
- class codar.savanna.runners.SummitRunner

Namespaces

- codar.savanna.runners

Variables

- codar.savanna.runners.mpiexec = MPIRunner('mpiexec', '-n', hostfile='--hostfile')
- codar.savanna.runners.aprun = MPIRunner('aprun', '-n', tasks_per_node_arg='-N', hostfile='-L')
- codar.savanna.runners.srun = MPIRunner('srun', '-n', nodes_arg='-N', hostfile='-w')
- codar.savanna.runners.jsrun = SummitRunner()

7.10 scheduler.py File Reference

Classes

- class codar.savanna.scheduler.JobList

Namespaces

- codar.savanna.scheduler

7.11 status.py File Reference

Classes

- class codar.savanna.status.WorkflowStatus
- class codar.savanna.status.PipelineState

Namespaces

- codar.savanna.status

Variables

- string codar.savanna.status.NOT_STARTED = 'not_started'
- string codar.savanna.status.RUNNING = 'running'
- string codar.savanna.status.DONE = 'done'
- string codar.savanna.status.KILLED = 'killed'
- string codar.savanna.status.REASON_TIMEOUT = 'timeout'
- string codar.savanna.status.REASON_FAILED = 'failed'
- string codar.savanna.status.REASON_SUCCEEDED = 'succeeded'
- string codar.savanna.status.REASON_EXCEPTION = 'exception'
- string codar.savanna.status.REASON_NOFIT = 'nofit'

7.12 summit_helper.py File Reference

Namespaces

- codar.savanna.summit_helper

Functions

- def codar.savanna.summit_helper.get_nodes_reqd (res_set, nrs)
- def codar.savanna.summit_helper.create_erf_file (run)

Index

__init__
    codar::savanna::consumer::PipelineRunner, 50
    codar::savanna::exc::MachineNotFound, 32
    codar::savanna::machines::Machine, 26
    codar::savanna::machines::MachineNode, 29
    codar::savanna::machines::SummitNode, 71
    codar::savanna::model::NodeConfig, 35
    codar::savanna::model::Pipeline, 42
    codar::savanna::model::Run, 59
    codar::savanna::node_layout::NodeLayout, 38
    codar::savanna::producer::JSONFilePipeline↩Reader, 24
    codar::savanna::runners::MPIRunner, 33
    codar::savanna::runners::SummitRunner, 73
    codar::savanna::scheduler::JobList, 22
    codar::savanna::status::PipelineState, 56
    codar::savanna::status::WorkflowStatus, 77
__init__.py, 79
__len__
    codar::savanna::scheduler::JobList, 22

add_callback
    codar::savanna::model::Run, 59
add_done_callback
    codar::savanna::model::Pipeline, 43
add_fatal_callback
    codar::savanna::model::Pipeline, 43
add_job
    codar::savanna::scheduler::JobList, 22
add_node
    codar::savanna::node_layout::NodeLayout, 38
add_pipeline
    codar::savanna::consumer::PipelineRunner, 50
allocated_nodes
    codar::savanna::consumer::PipelineRunner, 52
aprun
    codar::savanna::runners, 16
args
    codar::savanna::model::Run, 63
as_data
    codar::savanna::status::PipelineState, 56
as_data_list
    codar::savanna::node_layout::NodeLayout, 38

bind_arg
    codar::savanna::runners::SummitRunner, 74

callbacks
    codar::savanna::model::Run, 63
close
    codar::savanna::model::Run, 59
codar, 9
codar.savanna, 9
codar.savanna.consumer, 10
codar.savanna.consumer.PipelineRunner, 49
codar.savanna.exc, 10
codar.savanna.exc.MachineNotFound, 31
codar.savanna.exc.SavannaException, 69
codar.savanna.machines, 10
codar.savanna.machines.Machine, 25
codar.savanna.machines.MachineNode, 29
codar.savanna.machines.SummitNode, 70
codar.savanna.main, 13
codar.savanna.model, 14
codar.savanna.model.NodeConfig, 35
codar.savanna.model.Pipeline, 41
codar.savanna.model.Run, 57
codar.savanna.node_layout, 16
codar.savanna.node_layout.NodeLayout, 36
codar.savanna.producer, 16
codar.savanna.producer.JSONFilePipelineReader, 23
codar.savanna.runners, 16
codar.savanna.runners.MPIRunner, 32
codar.savanna.runners.Runner, 68
codar.savanna.runners.SummitRunner, 72
codar.savanna.scheduler, 17
codar.savanna.scheduler.JobList, 21
codar.savanna.status, 17
codar.savanna.status.PipelineState, 55
codar.savanna.status.WorkflowStatus, 76
codar.savanna.summit_helper, 20
codar::savanna::consumer::PipelineRunner
    __init__, 50
    add_pipeline, 50
    allocated_nodes, 52
    free_cv, 52
    free_nodes, 53
    job_list, 53
    job_list_cv, 53
    kill_all, 51
    machine_name, 53
    max_nodes, 53
    pipeline_fatal, 51
    pipeline_finished, 51
    pipelines, 53
    pipelines_lock, 54
    ppn, 54
    run_finished, 51
    run_pipelines, 52

runner, 54
stop, 52
codar::savanna::exc::MachineNotFound
__init__, 32
codar::savanna::machines
cori, 11
get_by_name, 11
local, 11
SCHEDULER_OPTIONS, 11
summit, 12
theta, 12
titan, 12
codar::savanna::machines::Machine
__init__, 26
dataspaces_servers_per_node, 27
get_nodes_reqd, 27
get_scheduler_options, 27
name, 27
node_class, 28
node_exclusive, 28
processes_per_node, 28
runner_name, 28
scheduler_name, 28
scheduler_options, 28
codar::savanna::machines::MachineNode
__init__, 29
cpu, 30
gpu, 30
to_json, 30
validate_layout, 30
codar::savanna::machines::SummitNode
__init__, 71
to_json, 71
validate_layout, 71
codar::savanna::main
consumer, 13
get_job_id, 13
main, 13
parse_args, 13
codar::savanna::model
KILL_WAIT, 14
RETURN_NAME, 14
STDERR_NAME, 15
STDOUT_NAME, 15
WAIT_DELAY_GIVE_UP, 15
WAIT_DELAY_KILL, 15
WALLTIME_NAME, 15
codar::savanna::model::NodeConfig
__init__, 35
cpu, 35
gpu, 36
num_ranks_per_node, 36
codar::savanna::model::Pipeline
__init__, 42
add_done_callback, 43
add_fatal_callback, 43
done_callbacks, 46
fatal_callbacks, 46
force_kill_all, 43
from_data, 43
get_nodes_used, 44
get_pids, 44
get_state, 44
id, 46
join_all, 44
kill_on_partial_failure, 46
launch_mode, 47
log_prefix, 47
machine_name, 47
node_layout, 47
nodes_assigned, 47
post_process_args, 47
post_process_script, 48
post_process_stop_on_failure, 48
remove_done_callback, 44
remove_fatal_callback, 45
run_finished, 45
run_post_process_script, 45
runs, 48
set_ppn, 45
set_total_nodes, 45
start, 46
total_nodes, 48
total_procs, 48
working_dir, 48
codar::savanna::model::Run
__init__, 59
add_callback, 59
args, 63
callbacks, 63
close, 59
create_node_config, 60
depends_on_runs, 63
env, 63
erf_file, 64
exception, 60
exe, 64
from_data, 60
get_nodes_used, 60
get_pid, 61
get_returncode, 61
hostfile, 64
join, 61
kill, 61
killed, 61
log_prefix, 64
machine, 64
mpmd_run, 62
name, 64
node_config, 65
nodes, 65
nodes_assigned, 65
nprocs, 65
remove_callback, 62
res_set, 65
return_path, 65

run, 62
runner, 66
runner_override, 66
sched_args, 66
set_runner, 62
sleep_after, 66
stderr_path, 66
stdout_path, 66
succeeded, 62
tasks_per_node, 67
timed_out, 63
timeout, 67
walltime_path, 67
working_dir, 67
codar::savanna::node_layout::NodeLayout
__init__, 38
add_node, 38
as_data_list, 38
codes_per_node, 38
copy, 38
default_no_share_layout, 39
get_node_containing_code, 39
group_codes_by_node, 39
layout_list, 41
layout_map, 41
populate_remaining, 39
ppn, 40
serialize_to_dict, 40
shared_nodes, 40
validate, 40
codar::savanna::producer::JSONFilePipelineReader
__init__, 24
file_path, 25
read_pipelines, 25
codar::savanna::runners
aprun, 16
jsrun, 16
mpiexec, 17
srun, 17
codar::savanna::runners::MPIRunner
__init__, 33
exe, 34
hostfile, 34
nodes_arg, 34
nprocs_arg, 34
tasks_per_node_arg, 34
wrap, 34
codar::savanna::runners::Runner
wrap, 68
codar::savanna::runners::SummitRunner
__init__, 73
bind_arg, 74
cpus_per_rs_arg, 74
exe, 74
gpus_per_rs_arg, 74
launch_distribution_arg, 74
machine, 75
nrs_arg, 75
rs_per_host_arg, 75
tasks_per_rs_arg, 75
wrap, 73
wrap_deprecated, 73
codar::savanna::scheduler::JobList
__init__, 22
__len__, 22
add_job, 22
pop_job, 23
codar::savanna::status
DONE, 18
KILLED, 18
NOT_STARTED, 18
REASON_EXCEPTION, 18
REASON_FAILED, 19
REASON_NOFIT, 19
REASON_SUCCEEDED, 19
REASON_TIMEOUT, 19
RUNNING, 19
codar::savanna::status::PipelineState
__init__, 56
as_data, 56
id, 56
reason, 56
return_codes, 56
state, 57
codar::savanna::status::WorkflowStatus
__init__, 77
file_path, 77
set_state, 77
codar::savanna::summit_helper
create_erf_file, 20
get_nodes_reqd, 20
codes_per_node
codar::savanna::node_layout::NodeLayout, 38
consumer
codar::savanna::main, 13
consumer.py, 79
copy
codar::savanna::node_layout::NodeLayout, 38
cori
codar::savanna::machines, 11
cpu
codar::savanna::machines::MachineNode, 30
codar::savanna::model::NodeConfig, 35
cpus_per_rs_arg
codar::savanna::runners::SummitRunner, 74
create_erf_file
codar::savanna::summit_helper, 20
create_node_config
codar::savanna::model::Run, 60

DONE
codar::savanna::status, 18
dataspaces_servers_per_node
codar::savanna::machines::Machine, 27
default_no_share_layout
codar::savanna::node_layout::NodeLayout, 39
depends_on_runs

codar::savanna::model::Run, 63
done_callbacks
codar::savanna::model::Pipeline, 46

env
codar::savanna::model::Run, 63
erf_file
codar::savanna::model::Run, 64
exc.py, 79
exception
codar::savanna::model::Run, 60
exe
codar::savanna::model::Run, 64
codar::savanna::runners::MPIRunner, 34
codar::savanna::runners::SummitRunner, 74

fatal_callbacks
codar::savanna::model::Pipeline, 46
file_path
codar::savanna::producer::JSONFilePipeline↩Reader, 25
codar::savanna::status::WorkflowStatus, 77
force_kill_all
codar::savanna::model::Pipeline, 43
free_cv
codar::savanna::consumer::PipelineRunner, 52
free_nodes
codar::savanna::consumer::PipelineRunner, 53
from_data
codar::savanna::model::Pipeline, 43
codar::savanna::model::Run, 60

get_by_name
codar::savanna::machines, 11
get_job_id
codar::savanna::main, 13
get_node_containing_code
codar::savanna::node_layout::NodeLayout, 39
get_nodes_reqd
codar::savanna::machines::Machine, 27
codar::savanna::summit_helper, 20
get_nodes_used
codar::savanna::model::Pipeline, 44
codar::savanna::model::Run, 60
get_pid
codar::savanna::model::Run, 61
get_pids
codar::savanna::model::Pipeline, 44
get_returncode
codar::savanna::model::Run, 61
get_scheduler_options
codar::savanna::machines::Machine, 27
get_state
codar::savanna::model::Pipeline, 44
gpu
codar::savanna::machines::MachineNode, 30
codar::savanna::model::NodeConfig, 36
gpus_per_rs_arg
codar::savanna::runners::SummitRunner, 74
group_codes_by_node
codar::savanna::node_layout::NodeLayout, 39

hostfile
codar::savanna::model::Run, 64
codar::savanna::runners::MPIRunner, 34

id
codar::savanna::model::Pipeline, 46
codar::savanna::status::PipelineState, 56

job_list
codar::savanna::consumer::PipelineRunner, 53
job_list_cv
codar::savanna::consumer::PipelineRunner, 53
join
codar::savanna::model::Run, 61
join_all
codar::savanna::model::Pipeline, 44
jsrun
codar::savanna::runners, 16

KILL_WAIT
codar::savanna::model, 14
KILLED
codar::savanna::status, 18
kill
codar::savanna::model::Run, 61
kill_all
codar::savanna::consumer::PipelineRunner, 51
kill_on_partial_failure
codar::savanna::model::Pipeline, 46
killed
codar::savanna::model::Run, 61

launch_distribution_arg
codar::savanna::runners::SummitRunner, 74
launch_mode
codar::savanna::model::Pipeline, 47
layout_list
codar::savanna::node_layout::NodeLayout, 41
layout_map
codar::savanna::node_layout::NodeLayout, 41
local
codar::savanna::machines, 11
log_prefix
codar::savanna::model::Pipeline, 47
codar::savanna::model::Run, 64

machine
codar::savanna::model::Run, 64
codar::savanna::runners::SummitRunner, 75
machine_name
codar::savanna::consumer::PipelineRunner, 53
codar::savanna::model::Pipeline, 47
machines.py, 80
main
codar::savanna::main, 13
main.py, 80
max_nodes

codar::savanna::consumer::PipelineRunner, 53
model.py, 80
mpiexec
codar::savanna::runners, 17
mpmd_run
codar::savanna::model::Run, 62

NOT_STARTED
codar::savanna::status, 18
name
codar::savanna::machines::Machine, 27
codar::savanna::model::Run, 64
node_class
codar::savanna::machines::Machine, 28
node_config
codar::savanna::model::Run, 65
node_exclusive
codar::savanna::machines::Machine, 28
node_layout
codar::savanna::model::Pipeline, 47
node_layout.py, 81
nodes
codar::savanna::model::Run, 65
nodes_arg
codar::savanna::runners::MPIRunner, 34
nodes_assigned
codar::savanna::model::Pipeline, 47
codar::savanna::model::Run, 65
nprocs
codar::savanna::model::Run, 65
nprocs_arg
codar::savanna::runners::MPIRunner, 34
nrs_arg
codar::savanna::runners::SummitRunner, 75
num_ranks_per_node
codar::savanna::model::NodeConfig, 36

parse_args
codar::savanna::main, 13
pipeline_fatal
codar::savanna::consumer::PipelineRunner, 51
pipeline_finished
codar::savanna::consumer::PipelineRunner, 51
pipelines
codar::savanna::consumer::PipelineRunner, 53
pipelines_lock
codar::savanna::consumer::PipelineRunner, 54
pop_job
codar::savanna::scheduler::JobList, 23
populate_remaining
codar::savanna::node_layout::NodeLayout, 39
post_process_args
codar::savanna::model::Pipeline, 47
post_process_script
codar::savanna::model::Pipeline, 48
post_process_stop_on_failure
codar::savanna::model::Pipeline, 48
ppn
codar::savanna::consumer::PipelineRunner, 54
codar::savanna::node_layout::NodeLayout, 40
processes_per_node
codar::savanna::machines::Machine, 28
producer.py, 81

REASON_EXCEPTION
codar::savanna::status, 18
REASON_FAILED
codar::savanna::status, 19
REASON_NOFIT
codar::savanna::status, 19
REASON_SUCCEEDED
codar::savanna::status, 19
REASON_TIMEOUT
codar::savanna::status, 19
RETURN_NAME
codar::savanna::model, 14
RUNNING
codar::savanna::status, 19
read_pipelines
codar::savanna::producer::JSONFilePipeline↩
Reader, 25
reason
codar::savanna::status::PipelineState, 56
remove_callback
codar::savanna::model::Run, 62
remove_done_callback
codar::savanna::model::Pipeline, 44
remove_fatal_callback
codar::savanna::model::Pipeline, 45
res_set
codar::savanna::model::Run, 65
return_codes
codar::savanna::status::PipelineState, 56
return_path
codar::savanna::model::Run, 65
rs_per_host_arg
codar::savanna::runners::SummitRunner, 75
run
codar::savanna::model::Run, 62
run_finished
codar::savanna::consumer::PipelineRunner, 51
codar::savanna::model::Pipeline, 45
run_pipelines
codar::savanna::consumer::PipelineRunner, 52
run_post_process_script
codar::savanna::model::Pipeline, 45
runner
codar::savanna::consumer::PipelineRunner, 54
codar::savanna::model::Run, 66
runner_name
codar::savanna::machines::Machine, 28
runner_override
codar::savanna::model::Run, 66
runners.py, 81
runs
codar::savanna::model::Pipeline, 48

SCHEDULER_OPTIONS

codar::savanna::machines, 11
STDERR_NAME
codar::savanna::model, 15
STDOUT_NAME
codar::savanna::model, 15
sched_args
codar::savanna::model::Run, 66
scheduler.py, 82
scheduler_name
codar::savanna::machines::Machine, 28
scheduler_options
codar::savanna::machines::Machine, 28
serialize_to_dict
codar::savanna::node_layout::NodeLayout, 40
set_ppn
codar::savanna::model::Pipeline, 45
set_runner
codar::savanna::model::Run, 62
set_state
codar::savanna::status::WorkflowStatus, 77
set_total_nodes
codar::savanna::model::Pipeline, 45
shared_nodes
codar::savanna::node_layout::NodeLayout, 40
sleep_after
codar::savanna::model::Run, 66
srun
codar::savanna::runners, 17
start
codar::savanna::model::Pipeline, 46
state
codar::savanna::status::PipelineState, 57
status.py, 82
stderr_path
codar::savanna::model::Run, 66
stdout_path
codar::savanna::model::Run, 66
stop
codar::savanna::consumer::PipelineRunner, 52
succeeded
codar::savanna::model::Run, 62
summit
codar::savanna::machines, 12
summit_helper.py, 82

tasks_per_node
codar::savanna::model::Run, 67
tasks_per_node_arg
codar::savanna::runners::MPIRunner, 34
tasks_per_rs_arg
codar::savanna::runners::SummitRunner, 75
theta
codar::savanna::machines, 12
timed_out
codar::savanna::model::Run, 63
timeout
codar::savanna::model::Run, 67
titan
codar::savanna::machines, 12
to_json
codar::savanna::machines::MachineNode, 30
codar::savanna::machines::SummitNode, 71
total_nodes
codar::savanna::model::Pipeline, 48
total_procs
codar::savanna::model::Pipeline, 48

validate
codar::savanna::node_layout::NodeLayout, 40
validate_layout
codar::savanna::machines::MachineNode, 30
codar::savanna::machines::SummitNode, 71

WAIT_DELAY_GIVE_UP
codar::savanna::model, 15
WAIT_DELAY_KILL
codar::savanna::model, 15
WALLTIME_NAME
codar::savanna::model, 15
walltime_path
codar::savanna::model::Run, 67
working_dir
codar::savanna::model::Pipeline, 48
codar::savanna::model::Run, 67
wrap
codar::savanna::runners::MPIRunner, 34
codar::savanna::runners::Runner, 68
codar::savanna::runners::SummitRunner, 73
wrap_deprecated
codar::savanna::runners::SummitRunner, 73