Cheetah

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 Namespace Documentation **7**

4.1 codar.cheetah Namespace Reference 7

4.1.1 Detailed Description 7

4.2 codar.cheetah.adios2_interface Namespace Reference 7

4.2.1 Detailed Description 8

4.2.2 Function Documentation 8

4.2.2.1 get_adios_version() 8

4.2.2.2 set_engine() 8

4.2.2.3 set_transport() 8

4.2.2.4 set_var_operation() 9

4.3 codar.cheetah.adios_params Namespace Reference 9

4.3.1 Detailed Description 9

4.3.2 Function Documentation 9

4.3.2.1 adios_xml_transform() 9

4.3.2.2 xml_has_transport() 10

4.4 codar.cheetah.config Namespace Reference 10

4.4.1 Detailed Description 10
4.4.2 Function Documentation 10
4.4.2.1 get_dataspaces_num_servers() 10
4.5 codar.cheetah.exc Namespace Reference 11
4.5.1 Detailed Description 11
4.6 codar.cheetah.launchers Namespace Reference 11
4.6.1 Detailed Description 11
4.7 codar.cheetah.loader Namespace Reference 11
4.7.1 Detailed Description 11
4.7.2 Function Documentation 11
4.7.2.1 load_experiment_class() 12
4.8 codar.cheetah.model Namespace Reference 12
4.8.1 Detailed Description 12
4.9 codar.cheetah.parameters Namespace Reference 12
4.9.1 Detailed Description 13
4.10 codar.cheetah.pbs Namespace Reference 13
4.10.1 Detailed Description 13
4.10.2 Function Documentation 13
4.10.2.1 open_pbs_file() 13
4.10.2.2 write_run_script() 14
4.10.3 Variable Documentation 14
4.10.3.1 PBS_FORMAT_TEMPLATE 14
4.10.3.2 SUBMIT_FORMAT_TEMPLATE 14
4.11 codar.cheetah.report_generator Namespace Reference 14
4.11.1 Detailed Description 15
4.11.2 Function Documentation 15
4.11.2.1 generate_report() 15
4.12 codar.cheetah.runners Namespace Reference 15
4.12.1 Detailed Description 15
4.13 codar.cheetah.status Namespace Reference 15
4.13.1 Detailed Description 16
4.14 codar.cheetah.templates Namespace Reference 16
4.14.1 Detailed Description 16
4.14.2 Variable Documentation 16
4.14.2.1 CAMPAIGN_ENV_TEMPLATE 16
4.14.2.2 GROUP_ENV_TEMPLATE 16

5 Class Documentation 17

5.1 codar.cheetah.report_generator._ReportGenerator Class Reference 17

5.1.1 Detailed Description 17

5.1.2 Member Function Documentation 17

5.1.2.1 parse_campaign() 18

5.1.2.2 parse_run_dir() 18

5.1.2.3 parse_sweep_group() 18

5.1.2.4 parse_user_campaigns() 18

5.1.2.5 write_output() 18

5.2 codar.cheetah.report_generator._RunParser Class Reference 19

5.2.1 Constructor & Destructor Documentation 19

5.2.1.1 __init__() 19

5.2.2 Member Function Documentation 19

5.2.2.1 read_adios_output_file_sizes() 20

5.2.2.2 read_node_layout() 20

5.2.2.3 read_sos_perf_data() 20

5.2.2.4 serialize_params_nested_dict() 20

5.2.2.5 verify_run_successful() 21

5.3 codar.cheetah.model.Campaign Class Reference 21

5.3.1 Detailed Description 22

5.3.2 Member Function Documentation 22

5.3.2.1 make_experiment_run_dir() 22

5.4 codar.cheetah.exc.CampaignParseError Class Reference 23

5.5 codar.cheetah.exc.CheetahException Class Reference 24

5.6 codar.cheetah.parameters.CodeCommand Class Reference 25

5.6.1 Detailed Description 26

5.6.2 Member Function Documentation 26

5.6.2.1 add_arg() 26

5.7 codar.cheetah.parameters.Instance Class Reference 26

5.7.1 Detailed Description 27

5.7.2 Member Function Documentation 27
5.7.2.1 as_dict() 27
5.7.2.2 as_string() 28
5.7.2.3 code_commands() 28
5.7.2.4 get_codes_argv() 28
5.7.2.5 get_parameter_values_by_type() 28
5.7.2.6 parameter_values() 29
5.8 codar.cheetah.launchers.Launcher Class Reference 29
5.8.1 Detailed Description 30
5.8.2 Member Function Documentation 30
5.8.2.1 create_group_directory() 31
5.9 codar.cheetah.exc.MachineNotFound Class Reference 31
5.10 codar.cheetah.parameters.Param Class Reference 33
5.10.1 Detailed Description 34
5.11 codar.cheetah.parameters.ParamADIOS2XML Class Reference 35
5.11.1 Detailed Description 36
5.11.2 Constructor & Destructor Documentation 36
5.11.2.1 __init__() 36
5.12 codar.cheetah.parameters.ParamAdiosXML Class Reference 37
5.12.1 Detailed Description 38
5.13 codar.cheetah.parameters.ParamCmdLineArg Class Reference 38
5.13.1 Detailed Description 39
5.14 codar.cheetah.parameters.ParamCmdLineOption Class Reference 40
5.14.1 Detailed Description 41
5.15 codar.cheetah.parameters.ParamConfig Class Reference 41
5.15.1 Detailed Description 42
5.16 codar.cheetah.parameters.ParamEnvVar Class Reference 42
5.17 codar.cheetah.parameters.ParameterValue Class Reference 43
5.17.1 Detailed Description 44
5.18 codar.cheetah.parameters.ParamKeyValue Class Reference 45

5.18.1 Detailed Description 46
5.19 codar.cheetah.parameters.ParamRunner Class Reference 46
5.19.1 Detailed Description 47
5.20 codar.cheetah.parameters.ParamSchedulerArgs Class Reference 48
5.21 codar.cheetah.model.Run Class Reference 49
5.21.1 Detailed Description 50
5.21.2 Member Function Documentation 50
5.21.2.1 add_dataspaces_support() 50
5.21.2.2 get_app_param_dict() 50
5.21.2.3 insert_sosflow() 51
5.22 codar.cheetah.model.RunComponent Class Reference 51
5.23 codar.cheetah.runners.Runner Class Reference 52
5.23.1 Member Function Documentation 53
5.23.1.1 wrap_app_command() 53
5.24 codar.cheetah.runners.RunnerCray Class Reference 54
5.24.1 Member Function Documentation 55
5.24.1.1 wrap_app_command() 55
5.25 codar.cheetah.runners.RunnerLocal Class Reference 55
5.25.1 Member Function Documentation 56
5.25.1.1 wrap_app_command() 56
5.26 codar.cheetah.parameters.SummitOpts Class Reference 57
5.27 codar.cheetah.parameters.Sweep Class Reference 57
5.27.1 Detailed Description 58
5.27.2 Member Function Documentation 58
5.27.2.1 get_instances() 58
5.28 codar.cheetah.parameters.SweepGroup Class Reference 58
5.28.1 Detailed Description 59
5.29 codar.cheetah.parameters.SymLink Class Reference 60
5.29.1 Detailed Description 60

Index **61**

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

codar.cheetah 7
codar.cheetah.adios2_interface 7
codar.cheetah.adios_params 9
codar.cheetah.config 10
codar.cheetah.exc 11
codar.cheetah.launchers 11
codar.cheetah.loader 11
codar.cheetah.model 12
codar.cheetah.parameters 12
codar.cheetah.pbs 13
codar.cheetah.report_generator 14
codar.cheetah.runners 15
codar.cheetah.status 15
codar.cheetah.templates 16

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

codar.cheetah.report_generator._ReportGenerator 17
codar.cheetah.report_generator._RunParser 19
Exception
- codar.cheetah.exc.CheetahException 24
  - codar.cheetah.exc.CampaignParseError 23
  - codar.cheetah.exc.MachineNotFound 31

object
- codar.cheetah.launchers.Launcher 29
- codar.cheetah.model.Campaign 21
- codar.cheetah.model.Run 49
- codar.cheetah.model.RunComponent 51
- codar.cheetah.parameters.CodeCommand 25
- codar.cheetah.parameters.Instance 26
- codar.cheetah.parameters.Param 33
  - codar.cheetah.parameters.ParamADIOS2XML 35
  - codar.cheetah.parameters.ParamAdiosXML 37
  - codar.cheetah.parameters.ParamCmdLineArg 38
  - codar.cheetah.parameters.ParamCmdLineOption 40
  - codar.cheetah.parameters.ParamConfig 41
  - codar.cheetah.parameters.ParamEnvVar 42
  - codar.cheetah.parameters.ParamKeyValue 45
  - codar.cheetah.parameters.ParamRunner 46
  - codar.cheetah.parameters.ParamSchedulerArgs 48
- codar.cheetah.parameters.ParameterValue 43
- codar.cheetah.parameters.Sweep 57
- codar.cheetah.parameters.SweepGroup 58
- codar.cheetah.runners.Runner 52
  - codar.cheetah.runners.RunnerCray 54
  - codar.cheetah.runners.RunnerLocal 55

str
- codar.cheetah.parameters.SymLink 60

codar.cheetah.parameters.SummitOpts 57

Chapter 3

Class Index

3.1 Class List

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

codar.cheetah.report_generator._ReportGenerator 17
codar.cheetah.report_generator._RunParser 19
codar.cheetah.model.Campaign 21
codar.cheetah.exc.CampaignParseError 23
codar.cheetah.exc.CheetahException 24
codar.cheetah.parameters.CodeCommand 25
codar.cheetah.parameters.Instance 26
codar.cheetah.launchers.Launcher 29
codar.cheetah.exc.MachineNotFound 31
codar.cheetah.parameters.Param 33
codar.cheetah.parameters.ParamADIOS2XML 35
codar.cheetah.parameters.ParamAdiosXML 37
codar.cheetah.parameters.ParamCmdLineArg 38
codar.cheetah.parameters.ParamCmdLineOption 40
codar.cheetah.parameters.ParamConfig 41
codar.cheetah.parameters.ParamEnvVar 42
codar.cheetah.parameters.ParameterValue 43
codar.cheetah.parameters.ParamKeyValue 45
codar.cheetah.parameters.ParamRunner 46
codar.cheetah.parameters.ParamSchedulerArgs 48
codar.cheetah.model.Run 49
codar.cheetah.model.RunComponent 51
codar.cheetah.runners.Runner 52
codar.cheetah.runners.RunnerCray 54
codar.cheetah.runners.RunnerLocal 55
codar.cheetah.parameters.SummitOpts 57
codar.cheetah.parameters.Sweep 57
codar.cheetah.parameters.SweepGroup 58
codar.cheetah.parameters.SymLink 60

Chapter 4

Namespace Documentation

4.1 codar.cheetah Namespace Reference

Namespaces

- adios2_interface
- adios_params
- config
- exc
- launchers
- loader
- model
- parameters
- pbs
- report_generator
- runners
- status
- templates

4.1.1 Detailed Description

```
Import most important classes into top level cheetah module namespace.
```

4.2 codar.cheetah.adios2_interface Namespace Reference

Functions

- def get_adios_version (xml_file)
- def set_engine (xmlfile, io_obj, engine_type, parameters=None)
- def set_transport (xmlfile, io_obj, transport_type, parameters=None)
- def set_var_operation (xmlfile, io_obj, var_name, operation, parameters=None)

4.2.1 Detailed Description

```
ADIOS2 Interface
```

4.2.2 Function Documentation

4.2.2.1 get_adios_version()

```
def codar.cheetah.adios2_interface.get_adios_version (
              xml_file )
```

```
Get the ADIOS version of this xml file.

:param xml_file: Path to the adios xml file
:return: 1 (adios version 1) or 2 (adios version 2)
```

4.2.2.2 set_engine()

```
def codar.cheetah.adios2_interface.set_engine (
              xmlfile,
              io_obj,
              engine_type,
              parameters = None )
```

```
Set the engine type for an input IO object.

:param xmlfile: String. The ADIOS2 xml file to be modified
:param io_obj: String. Name of the io object which contains the engine
:param engine_type: String. The engine type to be set for the io object
:param parameters: List. A list of dicts containing ’key and ’value’ keys
:return: True on success, False on error
```

4.2.2.3 set_transport()

```
def codar.cheetah.adios2_interface.set_transport (
              xmlfile,
              io_obj,
              transport_type,
              parameters = None )
```

```
Set the transport type for an io object

:param xmlfile: String. The ADIOS2 xml file to be modified
:param io_obj: String. Name of the io object that contains the engine
:param transport_type String. The transport type for this io object
:param parameters: A dict containing the parameter keys and values
:return: True on success, False on error
```

4.2.2.4 set_var_operation()

```
def codar.cheetah.adios2_interface.set_var_operation (
              xmlfile,
              io_obj,
              var_name,
              operation,
              parameters = None )
```

```
Set an operation on a variable

:param xmlfile: String. The ADIOS2 xml file to be modified
:param io_obj: String. Name of the io object that contains the engine
:param var_name String. Name of the variable
:param operation String. The operation to be performed on the variable
:param parameters: A dict containing the parameter keys and values
:return: True on success, False on error
```

4.3 codar.cheetah.adios_params Namespace Reference

Functions

- def adios_xml_transform (xml_filepath, group_name, var_name, value)
- def **adios_xml_transport** (xml_filepath, group_name, method_name, method_opts)
- def xml_has_transport (xml_filepath, transport_type)

4.3.1 Detailed Description

```
Functions for parsing and editing the ADIOS xml file to enable variable
transforms.  Transforms include compression and reduction. 'Transform'
is an ADIOS specific term.
```

4.3.2 Function Documentation

4.3.2.1 adios_xml_transform()

```
def codar.cheetah.adios_params.adios_xml_transform (
              xml_filepath,
              group_name,
              var_name,
              value )
```

```
Edit the ADIOS XML file to enable transform (compression/reduction) for a
variable

:param group_name:   Name of the variable that will be transformed
:param var_name:     Name of the variable that will be transformed
:param value:        Transform type and options (sz, zfp etc.)
:param xml_filepath: Aboslute path of the adios xml file. This will be in
                     the run directory.

TODO: add error handling, tests, e.g. for when variable not found in XML
file
```

4.3.2.2 xml_has_transport()

```
def codar.cheetah.adios_params.xml_has_transport (
              xml_filepath,
              transport_type )
```

```
Check ADIOS XML file if provided transport method is present for any group
:param xml_filepath: Path to the adios xml file
:param transport_type: Transport type (POSIX, MPI, MPI_AGGREGATE,
DATASPACES, DIMES, FLEXPATH)
:return: True if transport type found, else false.
```

4.4 codar.cheetah.config Namespace Reference

Functions

- def **scheduler_path** (scheduler_name)
- def **machine_submit_env_path** (machine_name)
- def **etc_path** (conf_name)
- def get_dataspaces_num_servers (num_dimes_clients, num_dataspaces_clients)

Variables

- **PACKAGE_PATH** = os.path.realpath(os.path.dirname(__file__))
- **DATA_PATH** = os.path.join(PACKAGE_PATH, "data")
- **CODAR_PATH** = os.path.realpath(os.path.join(PACKAGE_PATH, ".."))
- **CHEETAH_PATH_SCHEDULER** = os.path.join(DATA_PATH, "scheduler")
- **CHEETAH_PATH_MACHINE_CONFIG** = os.path.join(DATA_PATH, "machine_config")
- **WORKFLOW_SCRIPT** = os.path.join(CODAR_PATH, "savanna", "main.py")

4.4.1 Detailed Description

```
Cheetah paths and (in future) features for loading site configuration.
```

4.4.2 Function Documentation

4.4.2.1 get_dataspaces_num_servers()

```
def codar.cheetah.config.get_dataspaces_num_servers (
              num_dimes_clients,
              num_dataspaces_clients )
```

```
Get the number of dataspaces server instances that must be created for a
given number of client processes.
```

4.5 codar.cheetah.exc Namespace Reference

Classes

- class CampaignParseError
- class CheetahException
- class MachineNotFound

4.5.1 Detailed Description

```
Exceptions.
```

4.6 codar.cheetah.launchers Namespace Reference

Classes

- class Launcher

Variables

- string **TAU_PROFILE_PATTERN** = "codar.cheetah.tau-{code}"

4.6.1 Detailed Description

```
Class model for "launchers", which are responsible for taking an application
and mediating how it is run on a super computer or local machine. The only
supported launcher currently is swift-t. Swift allows us to configure how
each run within a sweep is parallelized, and handles details of submitting to
the correct scheduler and runner when passed appropriate options.
```

4.7 codar.cheetah.loader Namespace Reference

Functions

- def load_experiment_class (file_path)

4.7.1 Detailed Description

```
Functions for loading experiment python files by path.

Requires Python 3.3+
```

4.7.2 Function Documentation

4.7.2.1 load_experiment_class()

```
def codar.cheetah.loader.load_experiment_class (
              file_path )
```

```
Given the path to a python module containing an experiment, load the
module and find and return the class.
```

4.8 codar.cheetah.model Namespace Reference

Classes

- class Campaign
- class Run
- class RunComponent

Variables

- **RESERVED_CODE_NAMES** = set(['post-process'])

4.8.1 Detailed Description

```
Object oriented model to represent jobs to run on different Supercomputers or
workstations using different schedulers and runners (for running applications
on compute nodes from front end nodes), and allow pass through of scheduler
or runner specific options.

Subclasses representing specific types of schedulers, runners, and
supercomputers (machines) are specified in other modules with the corresponding
name.
```

4.9 codar.cheetah.parameters Namespace Reference

Classes

- class CodeCommand
- class Instance
- class Param
- class ParamADIOS2XML
- class ParamAdiosXML
- class ParamCmdLineArg
- class ParamCmdLineOption
- class ParamConfig
- class ParamEnvVar
- class ParameterValue
- class ParamKeyValue
- class ParamRunner
- class ParamSchedulerArgs
- class SummitOpts
- class Sweep
- class SweepGroup
- class SymLink

4.9.1 Detailed Description

```
Module containing classes for specifying paramter value sets and groupings
of parameters. Used in the Experiment specification in the 'runs' variable.
```

4.10 codar.cheetah.pbs Namespace Reference

Functions

- def open_pbs_file (scheduler_dir_path, name, project, nodes, walltime)
- def write_run_script (script_out_path, scheduler_dir_path)

Variables

- string **PBS_NAME** = 'job.pbs'
- string **PBS_FORMAT_TEMPLATE**
- string **SUBMIT_FORMAT_TEMPLATE**

4.10.1 Detailed Description

```
Module for generating PBS files for executing many jobs with the same
number of nodes.

TODO: define a common interface for schedulers, that works with at least
SLURM and PBS.

TODO: codify dir structure - scheduler dir contains scheduler script, has
subdir for each set of experiment parameters.
```

4.10.2 Function Documentation

4.10.2.1 open_pbs_file()

```
def codar.cheetah.pbs.open_pbs_file (
              scheduler_dir_path,
              name,
              project,
              nodes,
              walltime )
```

```
Open and write a PBS file to the specified path and return the open file
object for further writing. Caller is responsible for closing the file.

TODO: rather than passing back a file, this should probably return
an object with an 'add_run' function. There should also be a template
for the run output dir set somewhere - maybe other modules handle that,
it should not be scheduler specific.
```

4.10.2.2 write_run_script()

```
def codar.cheetah.pbs.write_run_script (
              script_out_path,
              scheduler_dir_path )
```

```
Write a bash script that will submit a PBS file generated by
`open_pbs_file` with the correct working directory and enironment.
This is the end user (experiment runner)'s entry point to start the
experiment.
```

4.10.3 Variable Documentation

4.10.3.1 PBS_FORMAT_TEMPLATE

```
string codar.cheetah.pbs.PBS_FORMAT_TEMPLATE
```

Initial value:

```
=  """
#!/bin/bash
#PBS -N {name}
#PBS -A {project}
#PBS -l nodes={nodes}
#PBS -l walltime={walltime}

"""
```

4.10.3.2 SUBMIT_FORMAT_TEMPLATE

```
string codar.cheetah.pbs.SUBMIT_FORMAT_TEMPLATE
```

Initial value:

```
=  """
#!/bin/bash

cd "{scheduler_directory}"
qsub {pbs_name}
"""
```

4.11 codar.cheetah.report_generator Namespace Reference

Classes

- class _ReportGenerator
- class _RunParser

Functions

- def generate_report (campaign_directory, user_run_script, output_file_path)

4.11.1 Detailed Description

```
Generate performance report from a completed campaign.
This module parses all run directories in all sweep groups to aggregate
information.
Runs sosflow analysis to collect data.

All parameters specified in the spec file must be used as column headers in
an output csv file.
```

4.11.2 Function Documentation

4.11.2.1 generate_report()

```
def codar.cheetah.report_generator.generate_report (
              campaign_directory,
              user_run_script,
              output_file_path )
```

```
This is a post-run function.
It walks the campaign tree and retrieves performance information
about all completed runs.
```

4.12 codar.cheetah.runners Namespace Reference

Classes

- class Runner
- class RunnerCray
- class RunnerLocal

4.12.1 Detailed Description

```
TODO: unused currently by SwiftLauncher, but may still be needed, so keeping
this module for now.
```

4.13 codar.cheetah.status Namespace Reference

Functions

- def **print_campaign_status** (campaign_directory, filter_user=None, filter_group=None, filter_run=None, filter_code=None, group_summary=False, run_summary=False, print_logs=False, log_level='DEBUG', return_codes=False, print_output=False, show_parameters=False)
- def **get_workflow_status** (status_file_path, print_counts=False, indent=0, print_return_codes=False, filter←_run=None, print_parameters=False, filter_code=None, run_summary=False, code_names=None)

4.13.1 Detailed Description

```
Funtions to print status information for campaigns.
```

4.14 codar.cheetah.templates Namespace Reference

Variables

- string **CAMPAIGN_ENV_TEMPLATE**
- string **GROUP_ENV_TEMPLATE**

4.14.1 Detailed Description

```
Templates for cheetah configuration. This should be used as little as possible:
ideally scripts should be stored separately and be independently testable.
For example, bash scripts can use environment variables for customization
instead of being templates.
```

4.14.2 Variable Documentation

4.14.2.1 CAMPAIGN_ENV_TEMPLATE

`string codar.cheetah.templates.CAMPAIGN_ENV_TEMPLATE`

Initial value:

```
=  """
export CODAR_CHEETAH_EXPERIMENT_DIR="{experiment_dir}"
export CODAR_CHEETAH_MACHINE_CONFIG="{machine_config}"
export CODAR_CHEETAH_APP_CONFIG="{app_config}"
export CODAR_WORKFLOW_SCRIPT="{workflow_script_path}"
export CODAR_WORKFLOW_RUNNER="{workflow_runner}"
export CODAR_CHEETAH_WORKFLOW_LOG_LEVEL="{workflow_debug_level}"
export CODAR_CHEETAH_UMASK="{umask}"
export CODAR_PYTHON="{codar_python}"
"""
```

4.14.2.2 GROUP_ENV_TEMPLATE

`string codar.cheetah.templates.GROUP_ENV_TEMPLATE`

Initial value:

```
=  """
export CODAR_CHEETAH_GROUP_WALLTIME="{walltime}"
export CODAR_CHEETAH_GROUP_MAX_PROCS="{max_procs}"

export CODAR_CHEETAH_SCHEDULER_ACCOUNT="{account}"
# queue on PBS, partition on SLURM
export CODAR_CHEETAH_SCHEDULER_QUEUE="{queue}"
# SLURM specific options
export CODAR_CHEETAH_SCHEDULER_CONSTRAINT="{constraint}"
export CODAR_CHEETAH_SCHEDULER_LICENSE="{license}"

export CODAR_CHEETAH_CAMPAIGN_NAME="{campaign_name}"

export CODAR_CHEETAH_GROUP_NAME="{group_name}"
export CODAR_CHEETAH_GROUP_NODES="{nodes}"
export CODAR_CHEETAH_GROUP_NODE_EXCLUSIVE="{node_exclusive}"
export CODAR_CHEETAH_GROUP_PROCESSES_PER_NODE="{processes_per_node}"
export CODAR_CHEETAH_MACHINE_NAME="{machine_name}"
"""
```

Chapter 5

Class Documentation

5.1 codar.cheetah.report_generator._ReportGenerator Class Reference

Public Member Functions

- def **__init__** (self, campaign_directory, user_run_script, output_filename)
- def parse_campaign (self)
- def parse_user_campaigns (self)
- def parse_sweep_group (self, group_dir)
- def parse_run_dir (self, run_dir, exit_status)
- def write_output (self)

Public Attributes

- **parsed_runs**
- **unique_keys**
- **campaign_directory**
- **user_run_script**
- **output_filename**
- **current_campaign_user**
- **run_status**

5.1.1 Detailed Description

5.1.2 Member Function Documentation

5.1.2.1 parse_campaign()

```
def codar.cheetah.report_generator._ReportGenerator.parse_campaign (
              self )
```

```
:return:
```

5.1.2.2 parse_run_dir()

```
def codar.cheetah.report_generator._ReportGenerator.parse_run_dir (
              self,
              run_dir,
              exit_status )
```

```
Parse run directory of a sweep group
```

5.1.2.3 parse_sweep_group()

```
def codar.cheetah.report_generator._ReportGenerator.parse_sweep_group (
              self,
              group_dir )
```

```
Parse sweep group and get post-run performance information
```

5.1.2.4 parse_user_campaigns()

```
def codar.cheetah.report_generator._ReportGenerator.parse_user_campaigns (
              self )
```

```
:return:
```

5.1.2.5 write_output()

```
def codar.cheetah.report_generator._ReportGenerator.write_output (
              self )
```

```
:return:
```

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

- report_generator.py

5.2 codar.cheetah.report_generator._RunParser Class Reference

Public Member Functions

- def __init__ (self, run_dir, exit_status, user_run_script)
- def **read_fob_json** (self)
- def **get_rc_names** (self)
- def **get_run_params** (self)
- def read_sos_perf_data (self)
- def **get_cheetah_perf_data** (self)
- def read_adios_output_file_sizes (self)
- def read_node_layout (self)
- def **execute_user_run_script** (self)
- def verify_run_successful (self)
- def serialize_params_nested_dict (self, nested_run_params_dict)

Public Attributes

- **run_dir**
- **exit_status**
- **user_run_script**
- **serialized_run_params**
- **fob_dict**
- **rc_names**
- **rc_working_dir**
- **rc_name_exe**

5.2.1 Constructor & Destructor Documentation

5.2.1.1 __init__()

```
def codar.cheetah.report_generator._RunParser.__init__ (
              self,
              run_dir,
              exit_status,
              user_run_script )
```

```
Class to parse a run directory.
:param run_dir:
```

5.2.2 Member Function Documentation

5.2.2.1 read_adios_output_file_sizes()

```
def codar.cheetah.report_generator._RunParser.read_adios_output_file_sizes (
              self )
```

```
:return:
```

5.2.2.2 read_node_layout()

```
def codar.cheetah.report_generator._RunParser.read_node_layout (
              self )
```

```
:return:
```

5.2.2.3 read_sos_perf_data()

```
def codar.cheetah.report_generator._RunParser.read_sos_perf_data (
              self )
```

```
:return: True if sos data was found, False otherwise
```

5.2.2.4 serialize_params_nested_dict()

```
def codar.cheetah.report_generator._RunParser.serialize_params_nested_dict (
              self,
              nested_run_params_dict )
```

```
codar.cheetah.run-params.json has the structure:
{
    app1: {
param1: value1
param2: value2
    }
    app2: {
param1: value1
param2: value2
    }
}

Serialize this structure so that we have
{app1__param1: value1, app1__param2:value2, and so on}.
```

5.2.2.5 verify_run_successful()

```
def codar.cheetah.report_generator._RunParser.verify_run_successful (
              self )
```

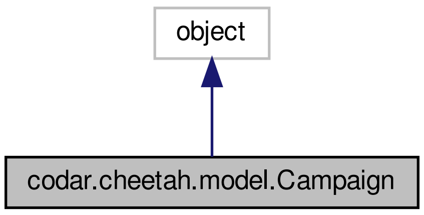
```
:return:
```

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

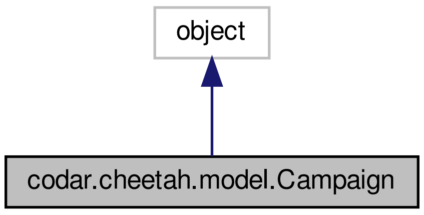
- report_generator.py

5.3 codar.cheetah.model.Campaign Class Reference

Inheritance diagram for codar.cheetah.model.Campaign:



Collaboration diagram for codar.cheetah.model.Campaign:



Public Member Functions

- def **__init__** (self, machine_name, app_dir)
- def make_experiment_run_dir (self, output_dir, _check_code_paths=True)

Public Attributes

- **machine**
- **app_dir**
- **runs**
- **inputs**
- **codes**
- **machine_scheduler_options**
- **machine_app_config_script**

Static Public Attributes

- **name** = None
- list **codes** = [ ]
- list **supported_machines** = [ ]
- list **sweeps** = [ ]
- list **inputs** = [ ]
- **umask** = None
- bool **kill_on_partial_failure** = False
- **run_post_process_script** = None
- bool **run_post_process_stop_group_on_failure** = False
- **app_config_scripts** = None
- **run_dir_setup_script** = None
- dictionary **scheduler_options** = {}
- **tau_config** = None
- **sosd_path** = None
- **sos_analysis_path** = None
- int **sosd_num_aggregators** = 1
- **post_process_script** = None
- **python_path** = sys.executable

5.3.1 Detailed Description

```
An experiment class specifies an application, a set of parameter to
sweep over, and a set of supported target machine. A specific instance
binds the experiment to a specific machine within the set of supported
machines, and supports generating a set of scripts to run the experiment
on that machine.
```

5.3.2 Member Function Documentation

5.3.2.1 make_experiment_run_dir()

```
def codar.cheetah.model.Campaign.make_experiment_run_dir (
              self,
              output_dir,
              _check_code_paths = True )
```

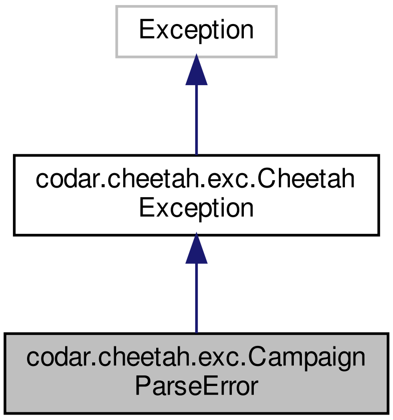
```
Produce scripts and directory structure for running the experiment.

Directory structure will be a subdirectory for each scheduler group,
and within each scheduler group directory, a subdirectory for each
run.
```

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

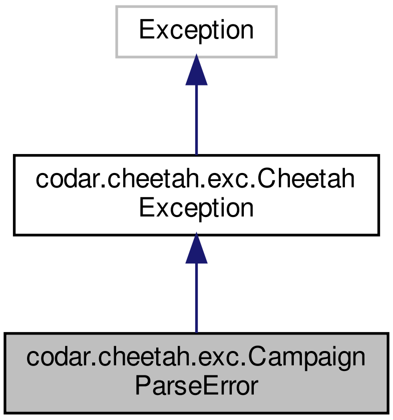
- model.py

5.4 codar.cheetah.exc.CampaignParseError Class Reference

Inheritance diagram for codar.cheetah.exc.CampaignParseError:


Exception
codar.cheetah.exc.Cheetah Exception
codar.cheetah.exc.Campaign ParseError

Collaboration diagram for codar.cheetah.exc.CampaignParseError:

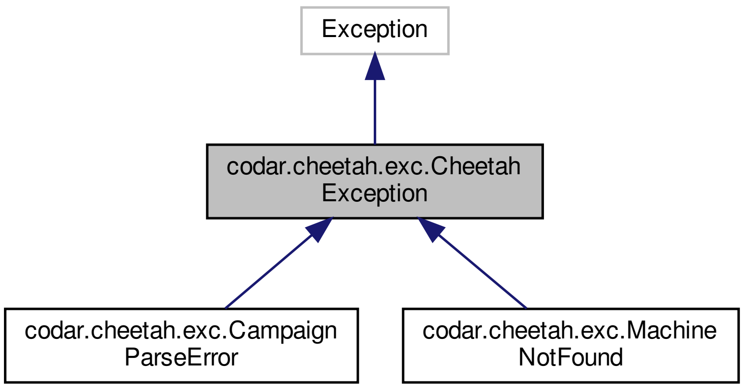

Exception
codar.cheetah.exc.Cheetah Exception
codar.cheetah.exc.Campaign ParseError

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

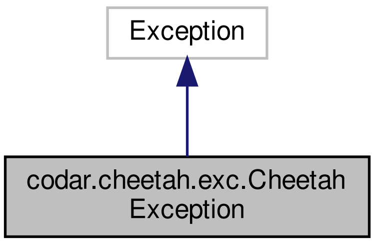
- exc.py

5.5 codar.cheetah.exc.CheetahException Class Reference

Inheritance diagram for codar.cheetah.exc.CheetahException:


Exception
codar.cheetah.exc.Cheetah Exception
codar.cheetah.exc.Campaign ParseError
codar.cheetah.exc.Machine NotFound

Collaboration diagram for codar.cheetah.exc.CheetahException:

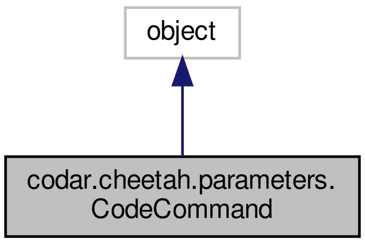


The documentation for this class was generated from the following file:

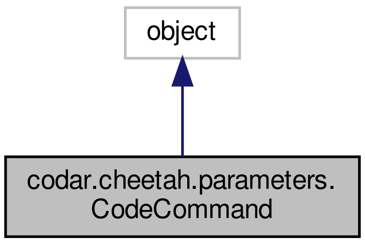
- exc.py

5.6 codar.cheetah.parameters.CodeCommand Class Reference

Inheritance diagram for codar.cheetah.parameters.CodeCommand:



Collaboration diagram for codar.cheetah.parameters.CodeCommand:



Public Member Functions

- def **__init__** (self, target)
- def add_arg (self, position, value)
- def **add_option** (self, option, value)
- def **get_argv** (self)

Public Attributes

- **target**
- **args**
- **options**

5.6.1 Detailed Description

```
Helper class to build up command args and options as we go. Does not
know about the path to it's executable, that is part of the execution
environment which is added during realization.
```

5.6.2 Member Function Documentation

5.6.2.1 add_arg()

```
def codar.cheetah.parameters.CodeCommand.add_arg (
              self,
              position,
              value )
```

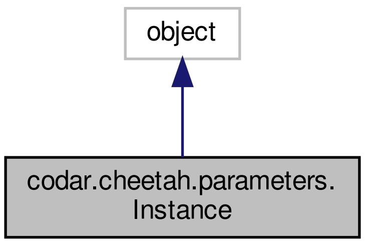
```
Allows adding positional args out of order.

TODO: better error handling.
```

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

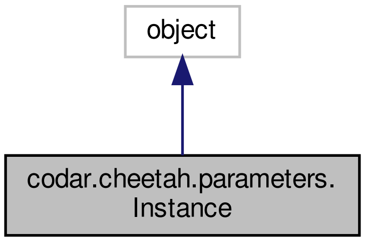
- parameters.py

5.7 codar.cheetah.parameters.Instance Class Reference

Inheritance diagram for codar.cheetah.parameters.Instance:



Collaboration diagram for codar.cheetah.parameters.Instance:



Public Member Functions

- def **__init__** (self)
- def **add_parameter** (self, p, idx)
- def parameter_values (self)
- def code_commands (self)
- def get_codes_argv (self)
- def as_string (self)
- def get_parameter_values_by_type (self, param_class)
- def **get_nprocs** (self, target)
- def **get_hostfile** (self, target)
- def **get_sched_opts** (self, target)
- def as_dict (self)

5.7.1 Detailed Description

```
Represent an instance of an application with fixed parameters. An
application may consistent of multiple codes running at the same time,
and multiple middlewear layers (scheduler like PBS, runner like aprun,
or swift), all of which may have their own parameters.

Abstractly, an instance is a two-level nested dict, where the first
level indicates the target for a parameter (application code or
middlewear), and the second level contains the parameter values for that
target.
```

5.7.2 Member Function Documentation

5.7.2.1 as_dict()

```
def codar.cheetah.parameters.Instance.as_dict (
              self )
```

```
Produce dict (mainly for for JSON seriliazation) with keys based on
parameter names. This ignores the type of the param, it's just the
name value pairs.
```

5.7.2.2 as_string()

```
def codar.cheetah.parameters.Instance.as_string (
              self )
```

```
Get a command line like value for the instance. Note that this
only includes positional and option command line args, not config
args like adios XML. TODO: deprecate??
```

5.7.2.3 code_commands()

```
def codar.cheetah.parameters.Instance.code_commands (
              self )
```

```
Wrapper to allow delayed calculation of derived parameter values.
```

5.7.2.4 get_codes_argv()

```
def codar.cheetah.parameters.Instance.get_codes_argv (
              self )
```

```
Get an _unordered_ dict mapping code name to list of args for
that code. Higher levels of model are responsible for re-ordering
as needed.
```

5.7.2.5 get_parameter_values_by_type()

```
def codar.cheetah.parameters.Instance.get_parameter_values_by_type (
              self,
              param_class )
```

```
Get a list of ParamaterValues of the specified type in the instance.
```

5.7.2.6 parameter_values()

```
def codar.cheetah.parameters.Instance.parameter_values (
              self )
```

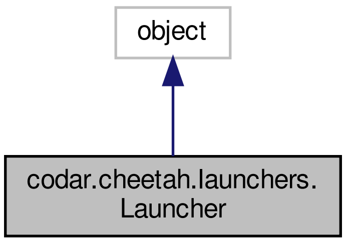
```
Wrapper to allow delayed calculation of derived parameter values.
```

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

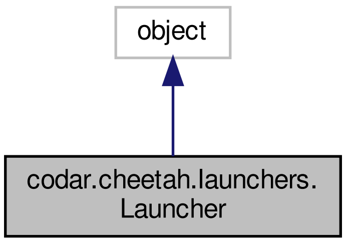
- parameters.py

5.8 codar.cheetah.launchers.Launcher Class Reference

Inheritance diagram for codar.cheetah.launchers.Launcher:



Collaboration diagram for codar.cheetah.launchers.Launcher:



Public Member Functions

- def **__init__** (self, machine_name, scheduler_name, runner_name, output_directory, num_codes)
- def create_group_directory (self, campaign_name, app_dir, group_name, runs, max_nprocs, nodes, launch↩_mode, component_subdirs, walltime, node_exclusive, timeout, machine, sosd_path=None, sos_analysis↩_path=None, tau_config=None, kill_on_partial_failure=False, run_post_process_script=None, run_post_↩process_stop_on_failure=False, scheduler_options=None, run_dir_setup_script=None)
- def **read_jobid** (self)

Public Attributes

- **machine_name**
- **scheduler_name**
- **runner_name**
- **output_directory**
- **num_codes**

Static Public Attributes

- **name** = None
- string **submit_script_name** = 'submit.sh'
- string **wait_script_name** = 'wait.sh'
- string **status_script_name** = 'status.sh'
- string **submit_out_name** = 'codar.cheetah.submit-output.txt'
- string **run_command_name** = 'codar.cheetah.run-params.txt'
- string **run_json_name** = 'codar.cheetah.run-params.json'
- string **run_out_name** = 'codar.cheetah.run-output.txt'
- **batch_script_name** = None
- string **batch_walltime_name** = 'codar.cheetah.walltime.txt'
- string **jobid_file_name** = 'codar.cheetah.jobid.txt'

5.8.1 Detailed Description

```
Class to represent a single batch job or submission script.
It’s job is to take a scheduler group and produce a script for executing
all runs within the scheduler group with the indicated scheduler
parameters.

The launcher may take configuration parameters to specify which scheduler/
runner to use, but there is no longer an object model for schedulers and
runners.
```

5.8.2 Member Function Documentation

5.8.2.1 create_group_directory()

```
def codar.cheetah.launchers.Launcher.create_group_directory (
              self,
              campaign_name,
              app_dir,
              group_name,
              runs,
              max_nprocs,
              nodes,
              launch_mode,
              component_subdirs,
              walltime,
              node_exclusive,
              timeout,
              machine,
              sosd_path = None,
              sos_analysis_path = None,
              tau_config = None,
              kill_on_partial_failure = False,
              run_post_process_script = None,
              run_post_process_stop_on_failure = False,
              scheduler_options = None,
              run_dir_setup_script = None )
```

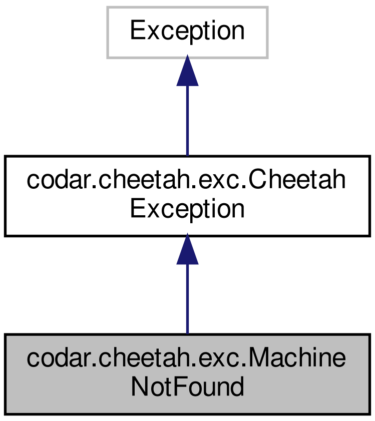
```
Copy scripts for the appropriate scheduler to group directory,
and write environment configuration. Returns required number of nodes,
which will be calculated if the passed nodes is None
```

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

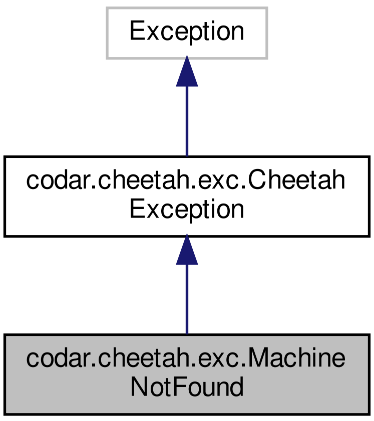
- launchers.py

5.9 codar.cheetah.exc.MachineNotFound Class Reference

Inheritance diagram for codar.cheetah.exc.MachineNotFound:


Exception
codar.cheetah.exc.Cheetah Exception
codar.cheetah.exc.Machine NotFound

Collaboration diagram for codar.cheetah.exc.MachineNotFound:


Exception
codar.cheetah.exc.Cheetah Exception
codar.cheetah.exc.Machine NotFound

Public Member Functions

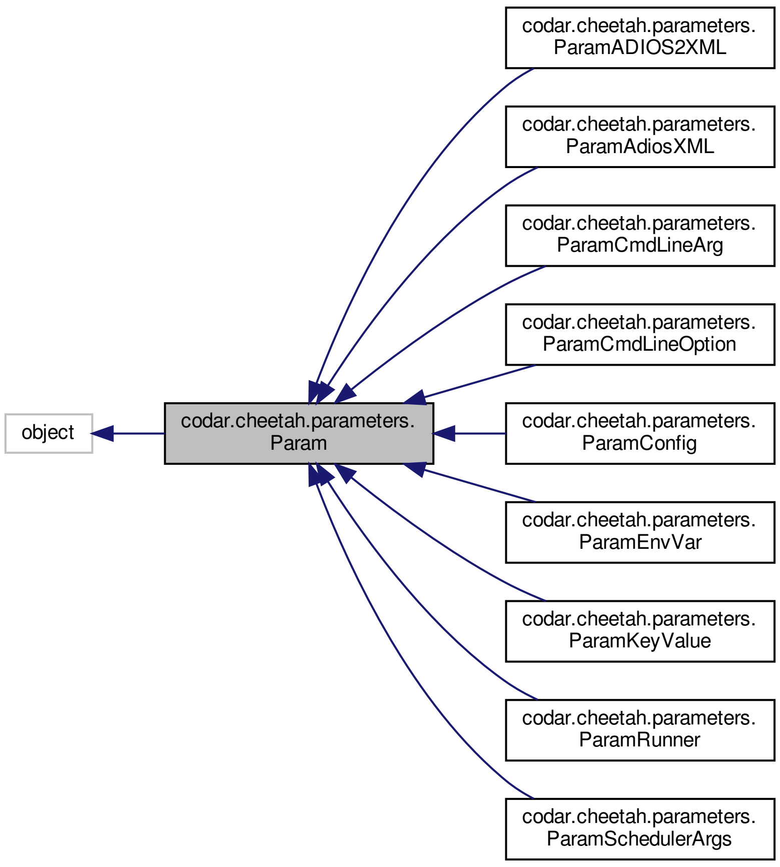
- def **__init__** (self, machine_name)

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

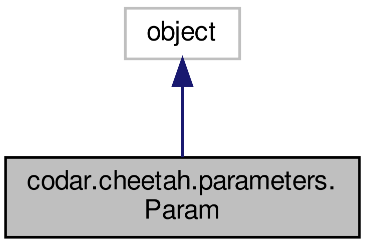
- exc.py

5.10 codar.cheetah.parameters.Param Class Reference

Inheritance diagram for codar.cheetah.parameters.Param:


codar.cheetah.parameters.ParamADIOS2XML
codar.cheetah.parameters.ParamAdiosXML
codar.cheetah.parameters.ParamCmdLineArg
codar.cheetah.parameters.ParamCmdLineOption
object
codar.cheetah.parameters.Param
codar.cheetah.parameters.ParamConfig
codar.cheetah.parameters.ParamEnvVar
codar.cheetah.parameters.ParamKeyValue
codar.cheetah.parameters.ParamRunner
codar.cheetah.parameters.ParamSchedulerArgs

Collaboration diagram for codar.cheetah.parameters.Param:



Public Member Functions

- def **__init__** (self, target, name, values)
- def **__get__** (self, idx)
- def **__len__** (self)

Public Attributes

- **target**
- **name**
- **values**

5.10.1 Detailed Description

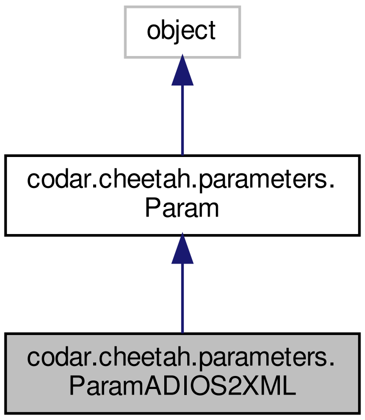
```
Abstract base class representing a parameter to an application. This
includes any method for modifying the run characteristics of an
application - command line, config file, environment variables, different
executable built with diffrent compiler flags.

Every parameter must have a unique name, and must target a specific
application or middleware, e.g. pbs, aprun, or one of the science
codes that make up an application.

Note that if a science application has only one code, it will likely still
involve middlewhere targets like PBS. Using a different target is one way
to model those.

TODO: is it useful to separate the definition of a param and it’s values?

TODO: should we require that the name be unique across all targets, or
just within each target? Global uniqueness allows for a simple list of
dict representation of instances, but two level nested dicts may be
more powerful (first level is target, second level is params).
```

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

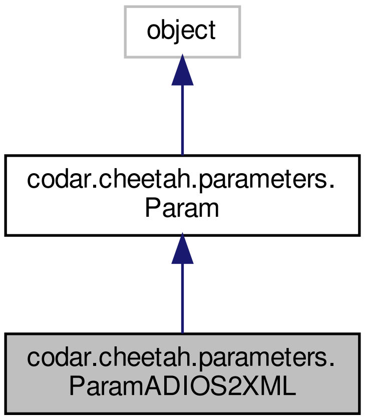
- parameters.py

5.11 codar.cheetah.parameters.ParamADIOS2XML Class Reference

Inheritance diagram for codar.cheetah.parameters.ParamADIOS2XML:


object
codar.cheetah.parameters.Param
codar.cheetah.parameters.ParamADIOS2XML

Collaboration diagram for codar.cheetah.parameters.ParamADIOS2XML:


object
codar.cheetah.parameters.Param
codar.cheetah.parameters.ParamADIOS2XML

Public Member Functions

- def __init__ (self, rc, io_name, operation_name, values)

Public Attributes

- **rc**
- **io_name**
- **operation_name**
- **values**

5.11.1 Detailed Description

```
Class to represent ADIOS2 XML file parameter options
```

5.11.2 Constructor & Destructor Documentation

5.11.2.1 __init__()

```
def codar.cheetah.parameters.ParamADIOS2XML.__init__ (
              self,
              rc,
              io_name,
              operation_name,
              values )
```

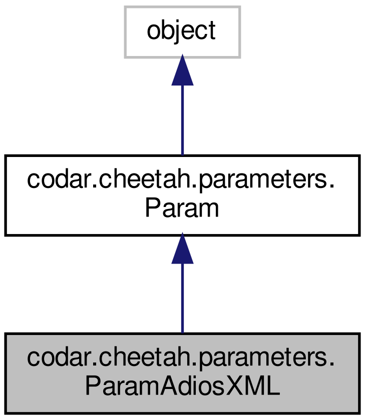
```
:param rc: name of the run component
:param io_name: name of the io object in the xml file
:param operation_name: engine/transport/var_operation
:param values: a list of dicts of the type
[ { engine_name: {parameters} },
  { engine_name: {parameters} },
  { var_name: {operation_name: {parameters}}}
]
Examples:
[ {"BPFile": {'Threads':1}},
  {"BPFile": {"ProfileUnits": "Microseconds"}}
]
[ { "T": { "zfp": {"rate":18, "accuracy": 0.01} } },
  { "T": { "zfp": {"rate":18, "accuracy": 0.001} } },
  { "T": { "zfp": {"rate":18, "accuracy": 0.0001} } },
  { "T": { "sz":  {"rate":18, "accuracy": 0.01} } },
]
```

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

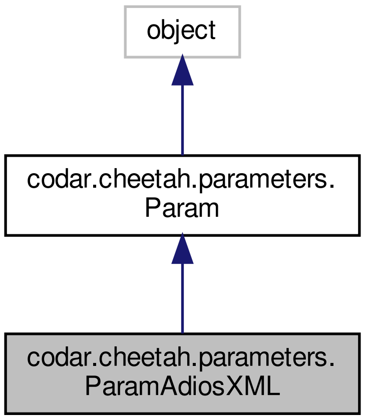
- parameters.py

5.12 codar.cheetah.parameters.ParamAdiosXML Class Reference

Inheritance diagram for codar.cheetah.parameters.ParamAdiosXML:



Collaboration diagram for codar.cheetah.parameters.ParamAdiosXML:



Public Member Functions

- def **__init__** (self, target, name, adios_xml_tags, values)

Public Attributes

- **param_type**
- **group_name**
- **var_name**

5.12.1 Detailed Description

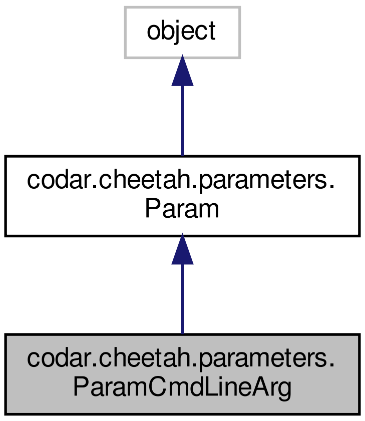
```
Class to represent ADIOS XML Transform.

The transform config is encoded in the name, so transforms on different
variables can be included in the sweep.

Format:
    adios_transform:<group_name>:<var_name>
    adios_transport:<group_name>

Note that the filename is specified in the code definition.
```

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

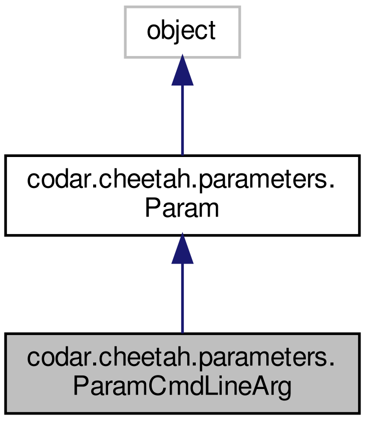
- parameters.py

5.13 codar.cheetah.parameters.ParamCmdLineArg Class Reference

Inheritance diagram for codar.cheetah.parameters.ParamCmdLineArg:


object
codar.cheetah.parameters.Param
codar.cheetah.parameters.ParamCmdLineArg

Collaboration diagram for codar.cheetah.parameters.ParamCmdLineArg:



Public Member Functions

- def **__init__** (self, target, name, position, values)

Public Attributes

- **position**

5.13.1 Detailed Description

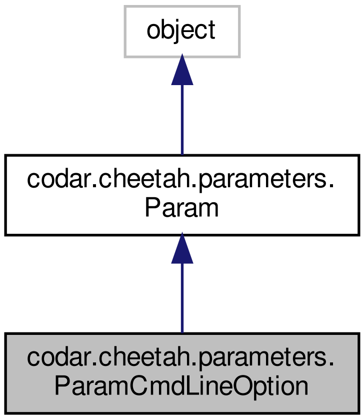
```
Specification for parameters that are based as a positional command line
argument.
```

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

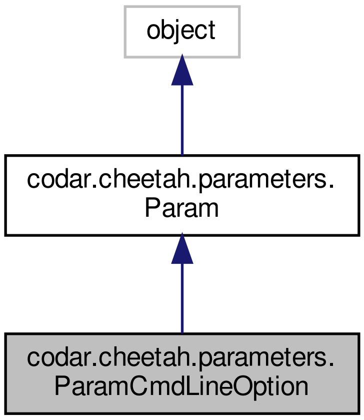
- parameters.py

5.14 codar.cheetah.parameters.ParamCmdLineOption Class Reference

Inheritance diagram for codar.cheetah.parameters.ParamCmdLineOption:



Collaboration diagram for codar.cheetah.parameters.ParamCmdLineOption:



Public Member Functions

- def **__init__** (self, target, name, option, values)

Public Attributes

- **option**

5.14.1 Detailed Description

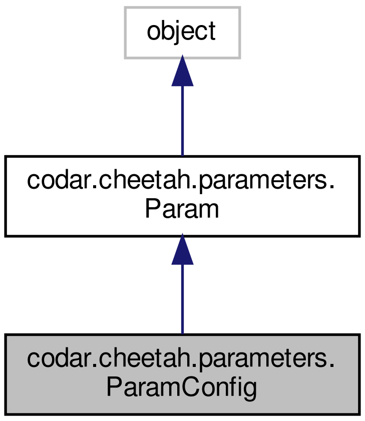
```
Specification for parameters that are based as a labeled command line
option. The option must contain the prefix, e.g. '--output-file' not
'output-file'.
```

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

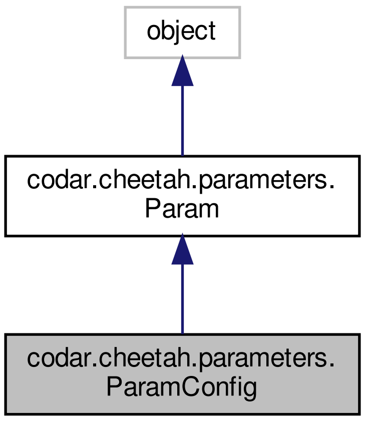
- parameters.py

5.15 codar.cheetah.parameters.ParamConfig Class Reference

Inheritance diagram for codar.cheetah.parameters.ParamConfig:



Collaboration diagram for codar.cheetah.parameters.ParamConfig:



Public Member Functions

- def **__init__** (self, target, name, config_filename, match_string, values)

Public Attributes

- **config_filename**
- **match_string**

5.15.1 Detailed Description

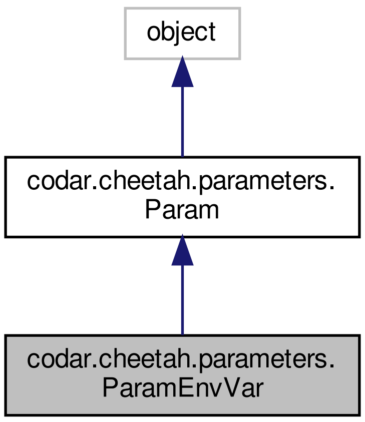
```
Class to represent a simple literal string replace in a config file.

Note that the filename must be added to the inputs list as well, to be
copied to each run directory.
```

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

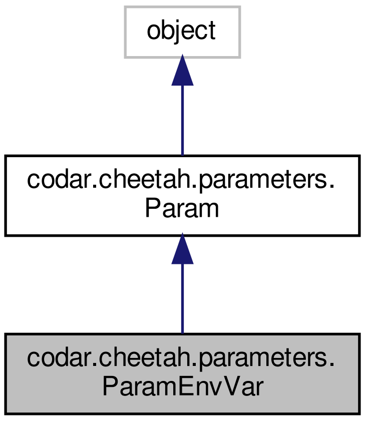
- parameters.py

5.16 codar.cheetah.parameters.ParamEnvVar Class Reference

Inheritance diagram for codar.cheetah.parameters.ParamEnvVar:



Collaboration diagram for codar.cheetah.parameters.ParamEnvVar:



Public Member Functions

- def **__init__** (self, target, name, option, values)

Public Attributes

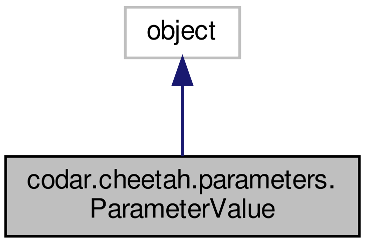
- **option**

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

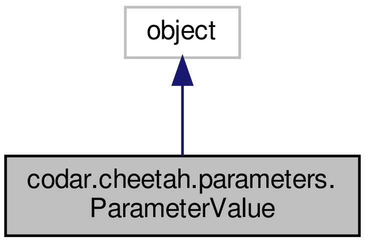
- parameters.py

5.17 codar.cheetah.parameters.ParameterValue Class Reference

Inheritance diagram for codar.cheetah.parameters.ParameterValue:



Collaboration diagram for codar.cheetah.parameters.ParameterValue:



Public Member Functions

- def **__init__** (self, parameter, value_index)
- def **__getattr__** (self, name)
- def **is_type** (self, parameter_class)

Public Attributes

- **value**

5.17.1 Detailed Description

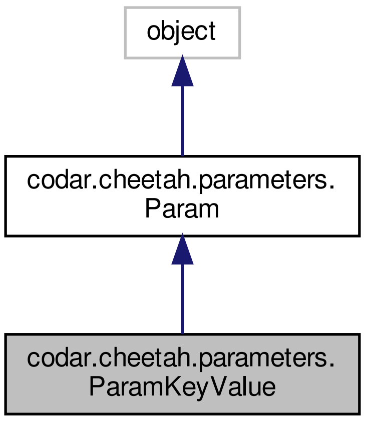
```
Convenience classes for tracking a specific value of a parameter.
Proxies to underlying parameter object, adds a `value` instance
variable.

TODO: this is kind of hacky, is there a better way?
```

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

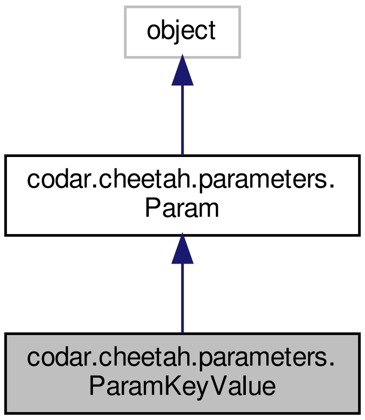
- parameters.py

5.18 codar.cheetah.parameters.ParamKeyValue Class Reference

Inheritance diagram for codar.cheetah.parameters.ParamKeyValue:



Collaboration diagram for codar.cheetah.parameters.ParamKeyValue:



Public Member Functions

- def **__init__** (self, target, name, config_filename, key_name, values)

Public Attributes

- **config_filename**
- **key_name**

5.18.1 Detailed Description

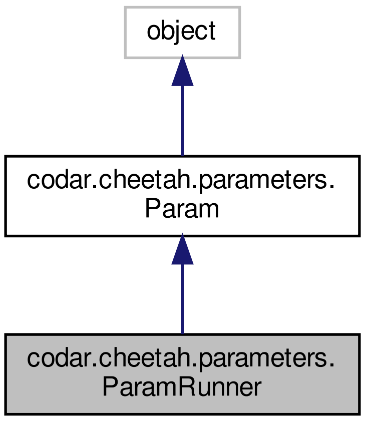
```
Class to represent replacement of the value in a config file with
'k = v' formatted lines. This should work with various formats, including
fortran namelist and INI, by ignoring lines that don't match the
simple k = v pattern. It has the advantage of being flexible, but the
disadvantage of not understanding sections or other more complicated
structure in config files. Also does not do any quoting - if required,
the spec writer should include literal quotes around the values.

Note that the filename must be added to the inputs list as well, to be
copied to each run directory.
```

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

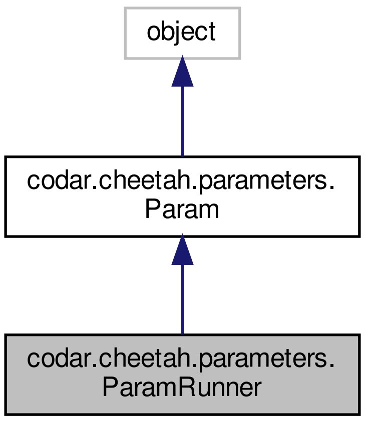
- parameters.py

5.19 codar.cheetah.parameters.ParamRunner Class Reference

Inheritance diagram for codar.cheetah.parameters.ParamRunner:


object
codar.cheetah.parameters.Param
codar.cheetah.parameters.ParamRunner

Collaboration diagram for codar.cheetah.parameters.ParamRunner:



Public Member Functions

- def **__init__** (self, target, name, values)

Additional Inherited Members

5.19.1 Detailed Description

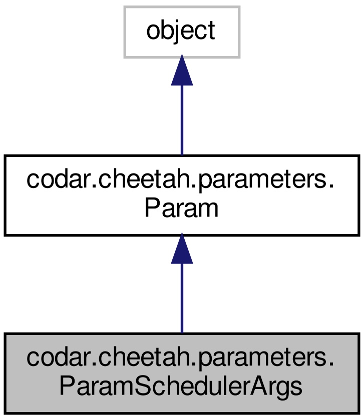
```
Specification for parameters that are passed to the runner, e.g.
mpirun, mpilaunch, srun, apirun, but usually still associated with a
specific application code.
```

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

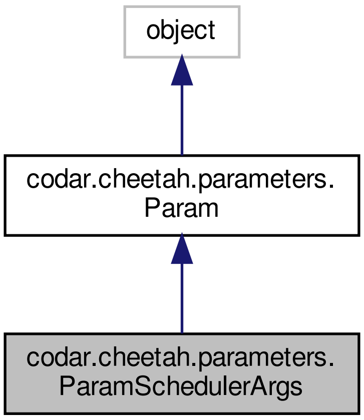
- parameters.py

5.20 codar.cheetah.parameters.ParamSchedulerArgs Class Reference

Inheritance diagram for codar.cheetah.parameters.ParamSchedulerArgs:



Collaboration diagram for codar.cheetah.parameters.ParamSchedulerArgs:



Public Member Functions

- def **__init__** (self, target, values)

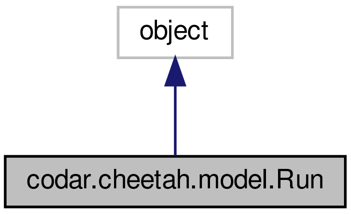
Additional Inherited Members

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

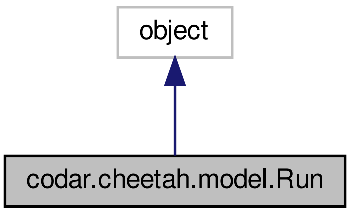
- parameters.py

5.21 codar.cheetah.model.Run Class Reference

Inheritance diagram for codar.cheetah.model.Run:



Collaboration diagram for codar.cheetah.model.Run:



Public Member Functions

- def **__init__** (self, instance, codes, codes_path, run_path, inputs, machine, node_layout, rc_dependency, component_subdirs, sosflow_profiling, sosflow_analyis, component_inputs=None)
- def **get_fob_data_list** (self)
- def **get_total_nprocs** (self)
- def get_app_param_dict (self)
- def add_dataspaces_support (self, machine)
- def insert_sosflow (self, sosd_path, sos_analysis_path, run_path, ppn)

Public Attributes

- **instance**
- **codes**
- **codes_path**
- **run_path**
- **run_id**
- **inputs**
- **machine**
- **node_layout**
- **component_subdirs**
- **sosflow_profiling**
- **sosflow_analysis**
- **component_inputs**
- **total_nodes**
- **run_components**

5.21.1 Detailed Description

```
Class representing how to actually run an instance on a given environment,
including how to generate arg arrays for executing each code required for
the application.

TODO: create a model shared between workflow and cheetah, i.e. codar.model
```

5.21.2 Member Function Documentation

5.21.2.1 add_dataspaces_support()

```
def codar.cheetah.model.Run.add_dataspaces_support (
              self,
              machine )
```

```
Add support for dataspaces.
Check RC Adios xml files to see if any transport methods are marked
for coupling with DATASPACES/DIMES.
For stage_write, check command line args to see if DATASPACES/DIMES
is specified.
:param machine: The current machine. I dont like this here.
:return:
```

5.21.2.2 get_app_param_dict()

```
def codar.cheetah.model.Run.get_app_param_dict (
              self )
```

```
Return dictionary containing only the app parameters
(does not include nprocs or exe paths).
```

5.21.2.3 insert_sosflow()

```
def codar.cheetah.model.Run.insert_sosflow (
              self,
              sosd_path,
              sos_analysis_path,
              run_path,
              ppn )
```

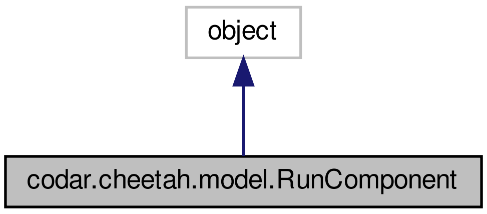
```
Insert a new component at start of list to launch sosflow daemon.
Should be called only once.
```

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

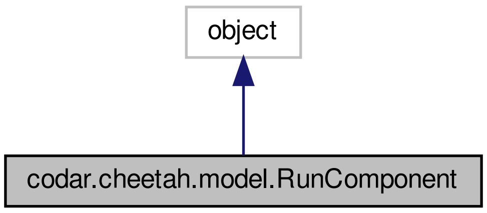
- model.py

5.22 codar.cheetah.model.RunComponent Class Reference

Inheritance diagram for codar.cheetah.model.RunComponent:



Collaboration diagram for codar.cheetah.model.RunComponent:



Public Member Functions

- def **__init__** (self, name, exe, args, sched_args, nprocs, working_dir, component_inputs=None, sleep↩ _after=None, linked_with_sosflow=False, adios_xml_file=None, env=None, timeout=None, hostfile=None, runner_override=False)
- def **as_fob_data** (self)

Public Attributes

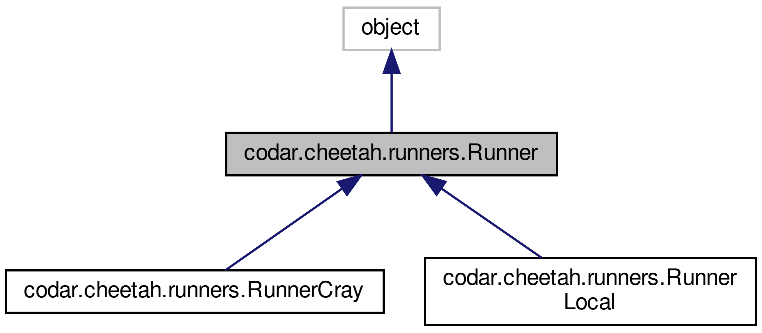
- **name**
- **exe**
- **args**
- **sched_args**
- **nprocs**
- **sleep_after**
- **env**
- **timeout**
- **working_dir**
- **component_inputs**
- **linked_with_sosflow**
- **adios_xml_file**
- **hostfile**
- **after_rc_done**
- **runner_override**

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

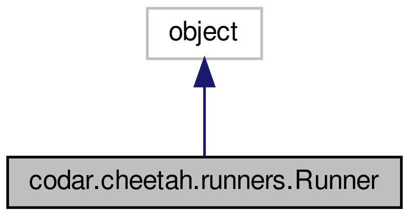
- model.py

5.23 codar.cheetah.runners.Runner Class Reference

Inheritance diagram for codar.cheetah.runners.Runner:


object
codar.cheetah.runners.Runner
codar.cheetah.runners.RunnerCray
codar.cheetah.runners.Runner Local

Collaboration diagram for codar.cheetah.runners.Runner:



Public Member Functions

- def wrap_app_command (self, command_dir, out_name, app_command)

Static Public Attributes

- **name** = None

5.23.1 Member Function Documentation

5.23.1.1 wrap_app_command()

```
def codar.cheetah.runners.Runner.wrap_app_command (
              self,
              command_dir,
              out_name,
              app_command )
```

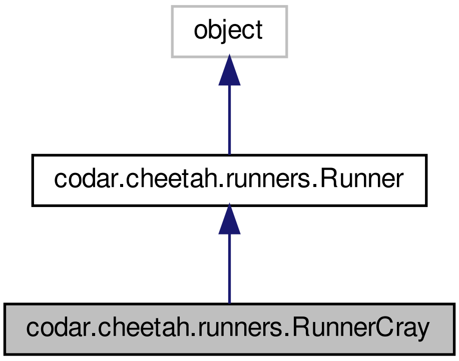
```
Given an application command line, return a list of commands to
run the given line using this runner and in the specified command
working directory.
```

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

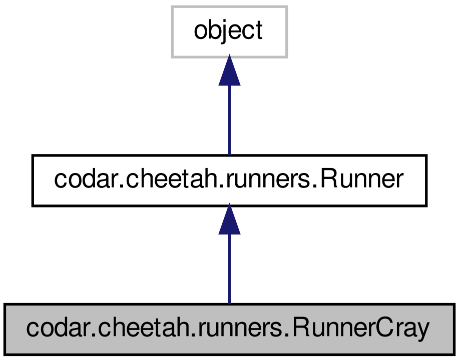
- runners.py

5.24 codar.cheetah.runners.RunnerCray Class Reference

Inheritance diagram for codar.cheetah.runners.RunnerCray:


object
codar.cheetah.runners.Runner
codar.cheetah.runners.RunnerCray

Collaboration diagram for codar.cheetah.runners.RunnerCray:


object
codar.cheetah.runners.Runner
codar.cheetah.runners.RunnerCray

Public Member Functions

- def wrap_app_command (self, command_dir, out_name, app_command)

Static Public Attributes

- string **name** = 'cray'

5.24.1 Member Function Documentation

5.24.1.1 wrap_app_command()

```
def codar.cheetah.runners.RunnerCray.wrap_app_command (
              self,
              command_dir,
              out_name,
              app_command )
```

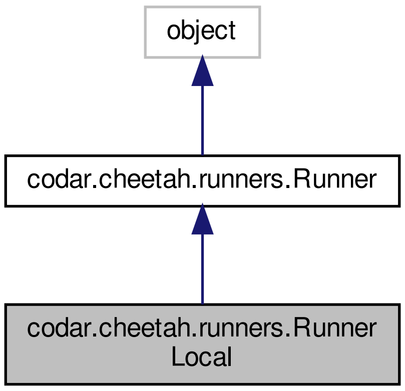
```
Run using aprun, and cd before/after to arrange separate working
dir per run.

TODO: how to pass aprun params?

NOTE: assumes CWD is batch directory within the experiment output dir.
```

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

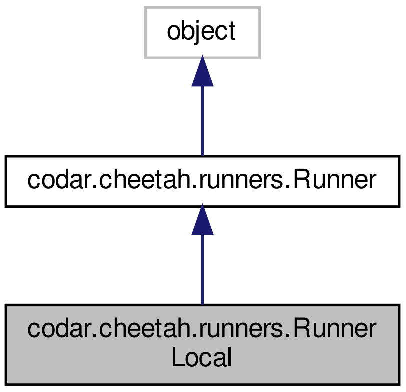
- runners.py

5.25 codar.cheetah.runners.RunnerLocal Class Reference

Inheritance diagram for codar.cheetah.runners.RunnerLocal:


object
codar.cheetah.runners.Runner
codar.cheetah.runners.Runner Local

Collaboration diagram for codar.cheetah.runners.RunnerLocal:


object
codar.cheetah.runners.Runner
codar.cheetah.runners.Runner Local

Public Member Functions

- def wrap_app_command (self, command_dir, out_name, app_command)

Static Public Attributes

- string **name** = 'local'

5.25.1 Member Function Documentation

5.25.1.1 wrap_app_command()

```
def codar.cheetah.runners.RunnerLocal.wrap_app_command (
              self,
              command_dir,
              out_name,
              app_command )
```

```
Run directly, just at cd before/after to arrange separate working
dir per run.

TODO: how to pass runner params?

NOTE: assumes CWD is batch directory within the experiment output dir.
```

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

- runners.py

5.26 codar.cheetah.parameters.SummitOpts Class Reference

Public Member Functions

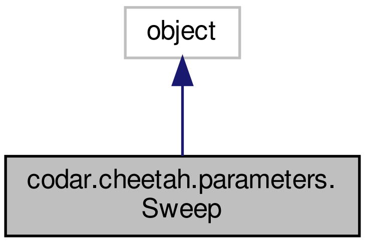
- def **__init__** (self)

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

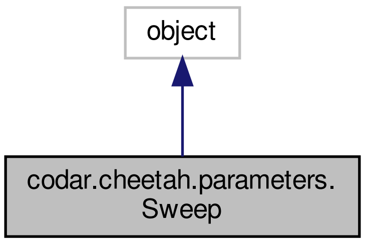
- parameters.py

5.27 codar.cheetah.parameters.Sweep Class Reference

Inheritance diagram for codar.cheetah.parameters.Sweep:



Collaboration diagram for codar.cheetah.parameters.Sweep:



Public Member Functions

- def **__init__** (self, parameters, node_layout=None, rc_dependency=None)
- def get_instances (self)

Public Attributes

- **parameters**
- **node_layout**
- **rc_dependency**

5.27.1 Detailed Description

```
Class representing a set of parameter values to search over as
a cross product.
```

5.27.2 Member Function Documentation

5.27.2.1 get_instances()

```
def codar.cheetah.parameters.Sweep.get_instances (
              self )
```

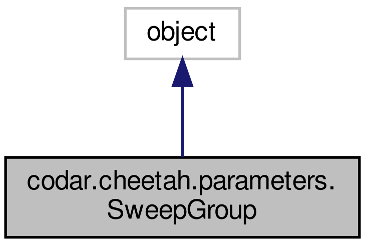
```
Get a list of Instance objects representing dense cross product over
param values.

TODO: this works great for command line options and args, but
what about for config and other types of params? Need to setup
a run dir and populate it with filled config templates.

Also how to pass per run output dir? Or is just making CWD the
per run dir enough for all cases we care about?

TODO: should have same signature as SweepGroup version OR a
different name.
```

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

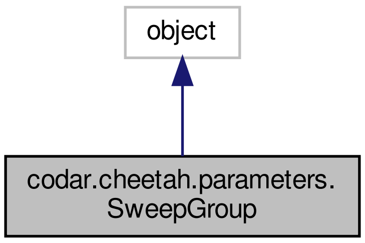
- parameters.py

5.28 codar.cheetah.parameters.SweepGroup Class Reference

Inheritance diagram for codar.cheetah.parameters.SweepGroup:



Collaboration diagram for codar.cheetah.parameters.SweepGroup:



Public Member Functions

- def **__init__** (self, name, parameter_groups, component_subdirs=False, component_inputs=None, walltime=3600, max_procs=None, per_run_timeout=None, sosflow_profiling=False, sosflow_analysis=False, nodes=None, launch_mode=None, run_repetitions=0)

Public Attributes

- **name**
- **nodes**
- **component_subdirs**
- **max_procs**
- **parameter_groups**
- **walltime**
- **per_run_timeout**
- **sosflow_profiling**
- **sosflow_analysis**
- **component_inputs**
- **launch_mode**
- **run_repetitions**

5.28.1 Detailed Description

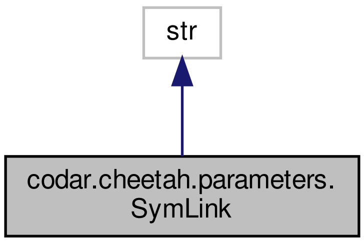
```
Class representing a grouping of run parameters that can be executed by
a single scheduler job, because they share the same scheduler parameters.

Note that nodes is no longer required - if not specified, it is calculated
based on the biggest run within the group.

How this gets converted into a script depends on the target machine and
which scheduler (if any) that machine uses.
```

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

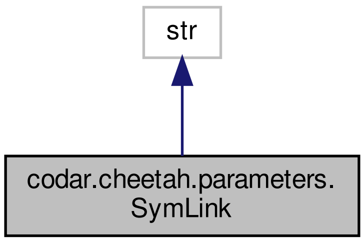
- parameters.py

5.29 codar.cheetah.parameters.SymLink Class Reference

Inheritance diagram for codar.cheetah.parameters.SymLink:



Collaboration diagram for codar.cheetah.parameters.SymLink:



Public Member Functions

- def **__init__** (self, source)

Public Attributes

- **source**

5.29.1 Detailed Description

```
Class to represent symbolic links as an input type for a run component
```

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

- parameters.py

Index

__init__
    codar::cheetah::parameters::ParamADIOS2XML, 36
    codar::cheetah::report_generator::_RunParser, 19

add_arg
    codar::cheetah::parameters::CodeCommand, 26
add_dataspaces_support
    codar::cheetah::model::Run, 50
adios_xml_transform
    codar::cheetah::adios_params, 9
as_dict
    codar::cheetah::parameters::Instance, 27
as_string
    codar::cheetah::parameters::Instance, 27

CAMPAIGN_ENV_TEMPLATE
    codar::cheetah::templates, 16
codar.cheetah, 7
codar.cheetah.adios2_interface, 7
codar.cheetah.adios_params, 9
codar.cheetah.config, 10
codar.cheetah.exc, 11
codar.cheetah.exc.CampaignParseError, 23
codar.cheetah.exc.CheetahException, 24
codar.cheetah.exc.MachineNotFound, 31
codar.cheetah.launchers, 11
codar.cheetah.launchers.Launcher, 29
codar.cheetah.loader, 11
codar.cheetah.model, 12
codar.cheetah.model.Campaign, 21
codar.cheetah.model.Run, 49
codar.cheetah.model.RunComponent, 51
codar.cheetah.parameters, 12
codar.cheetah.parameters.CodeCommand, 25
codar.cheetah.parameters.Instance, 26
codar.cheetah.parameters.Param, 33
codar.cheetah.parameters.ParamADIOS2XML, 35
codar.cheetah.parameters.ParamAdiosXML, 37
codar.cheetah.parameters.ParamCmdLineArg, 38
codar.cheetah.parameters.ParamCmdLineOption, 40
codar.cheetah.parameters.ParamConfig, 41
codar.cheetah.parameters.ParamEnvVar, 42
codar.cheetah.parameters.ParamKeyValue, 45
codar.cheetah.parameters.ParamRunner, 46
codar.cheetah.parameters.ParamSchedulerArgs, 48
codar.cheetah.parameters.ParameterValue, 43
codar.cheetah.parameters.SummitOpts, 57
codar.cheetah.parameters.Sweep, 57
codar.cheetah.parameters.SweepGroup, 58
codar.cheetah.parameters.SymLink, 60
codar.cheetah.pbs, 13
codar.cheetah.report_generator, 14
codar.cheetah.report_generator._ReportGenerator, 17
codar.cheetah.report_generator._RunParser, 19
codar.cheetah.runners, 15
codar.cheetah.runners.Runner, 52
codar.cheetah.runners.RunnerCray, 54
codar.cheetah.runners.RunnerLocal, 55
codar.cheetah.status, 15
codar.cheetah.templates, 16
codar::cheetah::adios2_interface
    get_adios_version, 8
    set_engine, 8
    set_transport, 8
    set_var_operation, 8
codar::cheetah::adios_params
    adios_xml_transform, 9
    xml_has_transport, 9
codar::cheetah::config
    get_dataspaces_num_servers, 10
codar::cheetah::launchers::Launcher
    create_group_directory, 30
codar::cheetah::loader
    load_experiment_class, 11
codar::cheetah::model::Campaign
    make_experiment_run_dir, 22
codar::cheetah::model::Run
    add_dataspaces_support, 50
    get_app_param_dict, 50
    insert_sosflow, 50
codar::cheetah::parameters::CodeCommand
    add_arg, 26
codar::cheetah::parameters::Instance
    as_dict, 27
    as_string, 27
    code_commands, 28
    get_codes_argv, 28
    get_parameter_values_by_type, 28
    parameter_values, 28
codar::cheetah::parameters::ParamADIOS2XML
    __init__, 36
codar::cheetah::parameters::Sweep
    get_instances, 58
codar::cheetah::pbs
    open_pbs_file, 13
    PBS_FORMAT_TEMPLATE, 14
    SUBMIT_FORMAT_TEMPLATE, 14
    write_run_script, 13

codar::cheetah::report_generator
  generate_report, 15
codar::cheetah::report_generator::_ReportGenerator
  parse_campaign, 17
  parse_run_dir, 18
  parse_sweep_group, 18
  parse_user_campaigns, 18
  write_output, 18
codar::cheetah::report_generator::_RunParser
  __init__, 19
  read_adios_output_file_sizes, 19
  read_node_layout, 20
  read_sos_perf_data, 20
  serialize_params_nested_dict, 20
  verify_run_successful, 20
codar::cheetah::runners::Runner
  wrap_app_command, 53
codar::cheetah::runners::RunnerCray
  wrap_app_command, 55
codar::cheetah::runners::RunnerLocal
  wrap_app_command, 56
codar::cheetah::templates
  CAMPAIGN_ENV_TEMPLATE, 16
  GROUP_ENV_TEMPLATE, 16
code_commands
  codar::cheetah::parameters::Instance, 28
create_group_directory
  codar::cheetah::launchers::Launcher, 30

GROUP_ENV_TEMPLATE
  codar::cheetah::templates, 16
generate_report
  codar::cheetah::report_generator, 15
get_adios_version
  codar::cheetah::adios2_interface, 8
get_app_param_dict
  codar::cheetah::model::Run, 50
get_codes_argv
  codar::cheetah::parameters::Instance, 28
get_dataspaces_num_servers
  codar::cheetah::config, 10
get_instances
  codar::cheetah::parameters::Sweep, 58
get_parameter_values_by_type
  codar::cheetah::parameters::Instance, 28

insert_sosflow
  codar::cheetah::model::Run, 50

load_experiment_class
  codar::cheetah::loader, 11

make_experiment_run_dir
  codar::cheetah::model::Campaign, 22

open_pbs_file
  codar::cheetah::pbs, 13

PBS_FORMAT_TEMPLATE
  codar::cheetah::pbs, 14

parameter_values
  codar::cheetah::parameters::Instance, 28
parse_campaign
  codar::cheetah::report_generator::_Report↩
    Generator, 17
parse_run_dir
  codar::cheetah::report_generator::_Report↩
    Generator, 18
parse_sweep_group
  codar::cheetah::report_generator::_Report↩
    Generator, 18
parse_user_campaigns
  codar::cheetah::report_generator::_Report↩
    Generator, 18

read_adios_output_file_sizes
  codar::cheetah::report_generator::_RunParser, 19
read_node_layout
  codar::cheetah::report_generator::_RunParser, 20
read_sos_perf_data
  codar::cheetah::report_generator::_RunParser, 20

SUBMIT_FORMAT_TEMPLATE
  codar::cheetah::pbs, 14
serialize_params_nested_dict
  codar::cheetah::report_generator::_RunParser, 20
set_engine
  codar::cheetah::adios2_interface, 8
set_transport
  codar::cheetah::adios2_interface, 8
set_var_operation
  codar::cheetah::adios2_interface, 8

verify_run_successful
  codar::cheetah::report_generator::_RunParser, 20

wrap_app_command
  codar::cheetah::runners::Runner, 53
  codar::cheetah::runners::RunnerCray, 55
  codar::cheetah::runners::RunnerLocal, 56
write_output
  codar::cheetah::report_generator::_Report↩
    Generator, 18
write_run_script
  codar::cheetah::pbs, 13

xml_has_transport
  codar::cheetah::adios_params, 9