```
Policy (Adaptive, Plottable)
C_TYPE = 'Policy'
C_NAME = '????'
C_BUFFER_CLS = SARSBuffer
_action_space
_observeration_space
_id
 __init___(p_observation_space:MSpace,
        p_action_space:MSpace,
        p_buffer_size=1,
        p_ada=True,
        p_logging=True)
get_observation_space()
get_action_space()
get_id()
set_id(p_id)
set_random_seed( p_seed=None )
compute_action(p_obs:State)
_adapt(p_args[0]:SARSElement)
          Agent (Policy)
                                                                    PreProStep
C_TYPE = 'Agent'
C_NAME = "
_policy
_name
_envmodel
_previous_observation
_previous_action
_planning_depth
_planning_width
  _init__(
p_policy:Policy,
p_envmodel:EnvModel=None,
p_action_planner:ActionPlanner=None
p_planning_depth=0,
p_name=",
p_id=0,
p_ada=True,
p_logging=True)
get_name()
set_name(p_name)
_extract_observation(p_state:State)
add_prepro_step(p_step:PreProStep)
set_random_seed( p_seed=None )
compute_action(p_state:State)
_adapt(p_args[0]:State,
      p_args[1]:Reward)
clear_buffer()
        MultiAgent (Agent)
C_TYPE = 'Multi Agent'
C_NAME = "
C_SUFFIX = '.cfg'
_agents
 _init__(p_name=",
       p_ada=1.0,
p_logging=True)
load(p_path, p_filename=None): bool
save(p_path, p_filename=None): bool
add\_agent(p\_agent,\,p\_weight=1.0)
get_agents()
_adapt(p_args[0]:State,
      p_args[1]:Reward)
set_random_seed( p_seed=None )
compute_action(p_state): action
clear_buffer()
                                                                                               Legend:
update_plot()
                                                                                                method()
```

Depends on Inherits from

method()

New/inherited method/attribute to be implemented

Method/attribute is already implemented

ActionPlanner (Log)

C_TYPE = 'Action Planner'

C_NAME = '????'

compute_action(
 p_state:State,

p_depth)

clear_action_path()

p_policy:Policy, p_envmodel:EnvModel,

_action_path