ParallelSimulations + scenarios: list(WaterDistributionNetworkScenarioSimulator) __init__(scenarios) + run_all(n_max_cpu_cores, max_memory_consumption)

ScenarioConfigurationParser

+ parse_config(): list(WaterDistributionNetworkScenarioSimulator)

epyt_flow.data.networks

+ load_hanoi(demand_profile, download_dir): WaterDistributionNetworkScenarioSimulator

+ load_ltown(demand_profile, download_dir): WaterDistributionNetworkScenarioSimulator

+ load_net1(download_dir): WaterDistributionNetworkScenarioSimulator

epyt_flow.data.scenarios

+ load_leakdb(scenario_id, download_dir): ScenarioResults

+ load_battledim(download_dir): ScenarioResults

ScenarioResults

+ senario_info: ScenarioInfo + sensor_readings

+ load(file_in) + to_numpy()

+ export(DataExport)

+ get_anomalous_time_points(): list(int)

ScenarioInfo

+ inp_file: str + msx_file: str + general_params: dict

+ sensor_config: SensorConfig + epanet_events: list(EpanetEvent)

+ sensor_events: list(SensorReadingEvent)

+ __init__(...)

+ add_demand_uncertainty(Uncertainty) + add_epanet_event(EpanetEvent)

+ epanet: EpyT-MSX

+ sensor_config: SensorConfig

+ __init__(inp_file, msx_file)

+ randomize_demands()

+ epanet_events: list(EpanetEvent) + sensor_events: list(SensorReadingEvent)

+ get_estimate_memory_consumption()

+ get_scenario_info(): ScenarioInfo

+ add_leakage(Leak)

+ add_sensor_reading_event(SensorReadingEvent)

+ add_sensor_fault(SensorFault)

+ add_sensor_uncertainty(SensorUncertainty)

+ add_simple_chlorine_injection(node_id, source_type, pattern)

+ set_general_parameters(demand_model, quality_type, time_steps, ...)

 $Water {\tt Distribution Network Scenario Simulator}$

+ run_simulation(DataExport, hyd_export=False)

+ run_advanced_quality_simulation(HydData, DataExport)

SensorConfig

+ pressure_sensors: list(int)

+ apply(...)

+ flow_sensors: list(int)

DataExport + __init__(file_out, output_info) + export(ScenarioInfo, sensor_readings) parent child child NumpyDataExport ExcelDataExport CsvDataExport MatlabDataExport

EpanetEvent + start_time: int + apply(EpyT-MSX) parent Leakage + location: int + diameter: float + profile: Pattern + __init__(location, start_time, end_time, diameter, profile) + apply(EpyT-MSX) AbruptLeakage IncipientLeakage peak_time: int + ___init___() __init__(peak_time)

SensorReadingEvent + start_time: int + end_time: int + sensor_id: int + sensor_type: int + __init__(...) + apply(SensorConfig, SensorReadings) SensorUncertainty SensorFault + uncertainty: Uncertainty + apply(SensorConfig, SensorReadings) + __init__(Uncertainty) + apply(SensorConfig, SensorReadings) StuckAtZero ConstantOffset Drift + param: float + param: float + apply(...)

+ apply(...)

ScenarioVisualizer + scenario: WaterDistributionNetworkScenarioSimulator + __init__(WaterDistributionNetworkScenarioSimulator) + plot_network_topology() + plot_animation(time_window)

epyt_flow.anomaly_detectors + get_linear_detector(): AnomalyDetector



