# The QualiMaster infrastructure – Specific rt-VIL types and commands

The QualiMaster rt-VIL binding library allows accessing specific types and operations provided by the QualiMaster infrastructure in rt-VIL to control the adaptation. This document summarizes the operations. In particular regarding the event types, please also consult the Javadoc documentation of the infrastructure. Notation follows the VIL conventions, so please consult the EASy-Producer / VIL documentation first.

## Infrastructure types

Infrastructure types such as relevant events and commands are mapped automatically into QM-IConf. See the development guidelines on the annotations on how to map further types or how to exclude individual types or methods from the mapping.

* AbstractCommandContainer / qualimaster::AbstractCommandContainer
  + Boolean keepOrdering(AbstractCommandContainer)
  + Boolean ==(Any,Any)
  + add(AbstractCommandContainer,CoordinationCommand)
  + Boolean !=(Any,Any)
  + CoordinationCommand command(AbstractCommandContainer,Integer)
  + Integer getCommandCount(AbstractCommandContainer)
  + Integer commandCount(AbstractCommandContainer)
  + Boolean <>(Any,Any)
  + CoordinationCommand getCommand(AbstractCommandContainer,Integer)
  + exec(AbstractCommandContainer)
  + new AbstractCommandContainer (sequenceOf)
  + new AbstractCommandContainer ()
* AbstractEvent / qualimaster::AbstractEvent
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
  + new AbstractEvent ()
* AbstractPipelineCommand / qualimaster::AbstractPipelineCommand
  + String pipeline(AbstractPipelineCommand)
  + exec(CoordinationCommand)
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + String getPipeline(AbstractPipelineCommand)
  + Boolean <>(Any,Any)
* AbstractPipelineElementCommand / qualimaster::AbstractPipelineElementCommand
  + String getPipelineElement(AbstractPipelineElementCommand)
  + String pipeline(AbstractPipelineCommand)
  + exec(CoordinationCommand)
  + Boolean ==(Any,Any)
  + String pipelineElement(AbstractPipelineElementCommand)
  + Boolean !=(Any,Any)
  + String getPipeline(AbstractPipelineCommand)
  + Boolean <>(Any,Any)
  + new AbstractPipelineElementCommand (String,String)
* AbstractPipelineElementEnactmentCompletedMonitoringEvent / qualimaster::AbstractPipelineElementEnactmentCompletedMonitoringEvent
  + String getPipelineElement(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String getPipeline(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String quoteNull(String)
  + String unquoteNull(String)
  + Boolean <>(Any,Any)
  + String causeMessageId(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String pipeline(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + Boolean ==(Any,Any)
  + Any key(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String pipelineElement(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + Boolean !=(Any,Any)
  + Any getKey(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String getCauseMessageId(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
* AdaptationEvent / qualimaster::AdaptationEvent
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
  + new AdaptationEvent ()
* AlgorithmChangeCommand / qualimaster::AlgorithmChangeCommand
  + PipelineOptions options(AlgorithmChangeCommand)
  + String getPipelineElement(AbstractPipelineElementCommand)
  + String algorithm(AlgorithmChangeCommand)
  + PipelineOptions getOptions(AlgorithmChangeCommand)
  + setIntParameter(AlgorithmChangeCommand,AlgorithmChangeParameter,Integer)
  + Integer intParameter(AlgorithmChangeCommand,AlgorithmChangeParameter,Integer)
  + String getStringParameter(AlgorithmChangeCommand,AlgorithmChangeParameter,String)
  + String getPipeline(AbstractPipelineCommand)
  + Boolean <>(Any,Any)
  + String getAlgorithm(AlgorithmChangeCommand)
  + String pipeline(AbstractPipelineCommand)
  + exec(CoordinationCommand)
  + String stringParameter(AlgorithmChangeCommand,AlgorithmChangeParameter,String)
  + Integer getIntParameter(AlgorithmChangeCommand,AlgorithmChangeParameter,Integer)
  + Boolean ==(Any,Any)
  + setStringParameter(AlgorithmChangeCommand,AlgorithmChangeParameter,String)
  + String pipelineElement(AbstractPipelineElementCommand)
  + Boolean !=(Any,Any)
  + setOptions(AlgorithmChangeCommand,PipelineOptions)
  + new AlgorithmChangeCommand (String,String,String)
* AlgorithmChangeParameter / qualimaster::AlgorithmChangeParameter
  + AlgorithmChangeParameter WARMUP\_DELAY
  + AlgorithmChangeParameter COPROCESSOR\_HOST
  + AlgorithmChangeParameter CONTROL\_RESPONSE\_PORT
  + AlgorithmChangeParameter CONTROL\_REQUEST\_PORT
  + AlgorithmChangeParameter INPUT\_PORT
  + AlgorithmChangeParameter OUTPUT\_PORT
  + AlgorithmChangeParameter IMPLEMENTING\_ARTIFACT
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* AlgorithmConfigurationAdaptationEvent / qualimaster::AlgorithmConfigurationAdaptationEvent
  + Boolean isUserTrigger(AlgorithmConfigurationAdaptationEvent)
  + String getAlgorithm(AlgorithmConfigurationAdaptationEvent)
  + String pipeline(AlgorithmConfigurationAdaptationEvent)
  + String getPipelineElement(AlgorithmConfigurationAdaptationEvent)
  + Boolean ==(Any,Any)
  + String pipelineElement(AlgorithmConfigurationAdaptationEvent)
  + String algorithm(AlgorithmConfigurationAdaptationEvent)
  + Boolean !=(Any,Any)
  + String getPipeline(AlgorithmConfigurationAdaptationEvent)
  + Boolean <>(Any,Any)
  + new AlgorithmConfigurationAdaptationEvent (String,String,String,Boolean)
* AnalysisObservables / qualimaster::AnalysisObservables
  + AnalysisObservables IS\_VALID
  + AnalysisObservables IS\_ENACTING
  + Boolean ==(Any,Any)
  + String name(AnalysisObservables)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* CheckBeforeStartupAdaptationEvent / qualimaster::CheckBeforeStartupAdaptationEvent
  + PipelineOptions options(CheckBeforeStartupAdaptationEvent)
  + String pipeline(CheckBeforeStartupAdaptationEvent)
  + Boolean ==(Any,Any)
  + PipelineOptions getOptions(CheckBeforeStartupAdaptationEvent)
  + Boolean !=(Any,Any)
  + String getPipeline(CheckBeforeStartupAdaptationEvent)
  + Boolean <>(Any,Any)
* CloudExecutionCommand / qualimaster::CloudExecutionCommand
  + Integer getNumVm(CloudExecutionCommand)
  + String getCloudEnvironment(CloudExecutionCommand)
  + Integer numVm(CloudExecutionCommand)
  + String cloudEnvironment(CloudExecutionCommand)
  + String getPipeline(CloudExecutionCommand)
  + status(CloudExecutionCommand)
  + Boolean <>(Any,Any)
  + exec(CloudExecutionCommand)
  + String pipeline(CloudExecutionCommand)
  + exec(CoordinationCommand)
  + Boolean ==(Any,Any)
  + accept(CloudExecutionCommand,PseudoVoid)
  + Boolean !=(Any,Any)
  + getStatus(CloudExecutionCommand)
  + new CloudExecutionCommand (String,String,Boolean)
  + new CloudExecutionCommand (String,String,Boolean,Integer)
* CloudResourceUsage / qualimaster::CloudResourceUsage
  + CloudResourceUsage PING
  + CloudResourceUsage BANDWIDTH
  + CloudResourceUsage USED\_HARDDISC\_MEM
  + CloudResourceUsage USED\_PROCESSORS
  + CloudResourceUsage USED\_WORKING\_STORAGE
  + Boolean ==(Any,Any)
  + String name(CloudResourceUsage)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* CommandSequence / qualimaster::CommandSequence
  + Boolean keepOrdering(CommandSequence)
  + exec(AbstractCommandContainer)
  + Boolean ==(Any,Any)
  + add(AbstractCommandContainer,CoordinationCommand)
  + Boolean !=(Any,Any)
  + CoordinationCommand command(AbstractCommandContainer,Integer)
  + Integer getCommandCount(AbstractCommandContainer)
  + Integer commandCount(AbstractCommandContainer)
  + Boolean <>(Any,Any)
  + CoordinationCommand getCommand(AbstractCommandContainer,Integer)
  + new CommandSequence (sequenceOf)
  + new CommandSequence ()
* CommandSet / qualimaster::CommandSet
  + Boolean keepOrdering(CommandSet)
  + exec(AbstractCommandContainer)
  + Boolean ==(Any,Any)
  + add(AbstractCommandContainer,CoordinationCommand)
  + Boolean !=(Any,Any)
  + CoordinationCommand command(AbstractCommandContainer,Integer)
  + Integer getCommandCount(AbstractCommandContainer)
  + Integer commandCount(AbstractCommandContainer)
  + Boolean <>(Any,Any)
  + CoordinationCommand getCommand(AbstractCommandContainer,Integer)
  + new CommandSet (sequenceOf)
  + new CommandSet ()
* ConstraintViolationAdaptationEvent / qualimaster::ConstraintViolationAdaptationEvent
  + Iterator(ViolatingClause) getViolatingClauses(ConstraintViolationAdaptationEvent)
  + Boolean ==(Any,Any)
  + ViolatingClause getViolatingClause(ConstraintViolationAdaptationEvent,Integer)
  + Integer getViolatingClauseCount(ConstraintViolationAdaptationEvent)
  + Boolean !=(Any,Any)
  + Iterator(ViolatingClause) violatingClauses(ConstraintViolationAdaptationEvent)
  + ViolatingClause violatingClause(ConstraintViolationAdaptationEvent,Integer)
  + Integer violatingClauseCount(ConstraintViolationAdaptationEvent)
  + Boolean <>(Any,Any)
* CoordinationCommand / qualimaster::CoordinationCommand
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
  + exec(CoordinationCommand)
  + new CoordinationCommand ()
* DefaultLoadShedders / qualimaster::DefaultLoadShedders
  + DefaultLoadShedders NO\_SHEDDING
  + DefaultLoadShedders NTH\_ITEM
  + DefaultLoadShedders PROBABILISTIC
  + DefaultLoadShedders FAIR\_PATTERN
  + Boolean ==(Any,Any)
  + String getIdentifier(DefaultLoadShedders)
  + String shortName(DefaultLoadShedders)
  + String getShortName(DefaultLoadShedders)
  + Boolean !=(Any,Any)
  + String identifier(DefaultLoadShedders)
  + Boolean <>(Any,Any)
* DefaultLoadSheddingParameter / qualimaster::DefaultLoadSheddingParameter
  + DefaultLoadSheddingParameter NTH\_TUPLE
  + DefaultLoadSheddingParameter PROBABILITY
  + DefaultLoadSheddingParameter RATIO
  + Boolean ==(Any,Any)
  + String name(DefaultLoadSheddingParameter)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* Event / qualimaster::Event
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* FrozenSystemState / qualimaster::FrozenSystemState
* String INFRASTRUCTURE
* String INFRASTRUCTURE\_NAME
* String MACHINE
* String HWNODE
* String CLOUDENV
* String ALGORITHM
* String DATASOURCE
* String DATASINK
* String PIPELINE
* String PIPELINE\_ELEMENT
* Real getInfrastructureObservation(FrozenSystemState,Observable)
* Real dataSourceObservation(FrozenSystemState,String,String,Observable)
* Real cloudObservation(FrozenSystemState,String,Observable,Real)
* Real getAlgorithmObservation(FrozenSystemState,String,String,Observable,Real)
* Real getDataSinkObservation(FrozenSystemState,String,String,Observable)
* Boolean <>(Any,Any)
* Real hwNodeObservation(FrozenSystemState,String,Observable)
* Real getDataSinkObservation(FrozenSystemState,String,String,Observable,Real)
* String obtainKey(String,String,Observable)
* Real getPipelineElementObservation(FrozenSystemState,String,String,Observable)
* Real getPipelineElementObservation(FrozenSystemState,String,String,Observable,Real)
* Boolean ==(Any,Any)
* Real getInfrastructureObservation(FrozenSystemState,Observable,Real)
* Boolean !=(Any,Any)
* Real infrastructureObservation(FrozenSystemState,Observable,Real)
* Real getMachineObservation(FrozenSystemState,String,Observable,Real)
* Real getCloudObservation(FrozenSystemState,String,Observable,Real)
* Real pipelineObservation(FrozenSystemState,String,Observable)
* Real getHwNodeObservation(FrozenSystemState,String,Observable)
* Real pipelineObservation(FrozenSystemState,String,Observable,Real)
* Real getMachineObservation(FrozenSystemState,String,Observable)
* Real getHwNodeObservation(FrozenSystemState,String,Observable,Real)
* Real dataSourceObservation(FrozenSystemState,String,String,Observable,Real)
* Real algorithmObservation(FrozenSystemState,String,String,Observable)
* Real algorithmObservation(FrozenSystemState,String,String,Observable,Real)
* Real getAlgorithmObservation(FrozenSystemState,String,String,Observable)
* Real infrastructureObservation(FrozenSystemState,Observable)
* Real getDataSourceObservation(FrozenSystemState,String,String,Observable,Real)
* Real dataSinkObservation(FrozenSystemState,String,String,Observable)
* Real pipelineElementObservation(FrozenSystemState,String,String,Observable)
* Real machineObservation(FrozenSystemState,String,Observable)
* Real hwNodeObservation(FrozenSystemState,String,Observable,Real)
* Real getDataSourceObservation(FrozenSystemState,String,String,Observable)
* Real cloudObservation(FrozenSystemState,String,Observable)
* Real pipelineElementObservation(FrozenSystemState,String,String,Observable,Real)
* Real dataSinkObservation(FrozenSystemState,String,String,Observable,Real)
* Real machineObservation(FrozenSystemState,String,Observable,Real)
* Real getPipelineObservation(FrozenSystemState,String,Observable,Real)
* Real getCloudObservation(FrozenSystemState,String,Observable)
* Real getPipelineObservation(FrozenSystemState,String,Observable)
* new FrozenSystemState (mapOf(String, Real))
* new FrozenSystemState ()
* FunctionalSuitability / qualimaster::FunctionalSuitability
  + FunctionalSuitability ACCURACY\_CONFIDENCE
  + FunctionalSuitability ACCURACY\_ERROR\_RATE
  + FunctionalSuitability BELIEVABILITY
  + FunctionalSuitability RELEVANCY
  + FunctionalSuitability COMPLETENESS
  + FunctionalSuitability NOVELTY
  + FunctionalSuitability DIVERSITY
  + FunctionalSuitability SERENDIPITY
  + FunctionalSuitability MP\_VOLATILITY
  + Boolean ==(Any,Any)
  + String name(FunctionalSuitability)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* InfrastructurePart / qualimaster::InfrastructurePart
  + InfrastructurePart SOFTWARE
  + InfrastructurePart HARDWARE
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* InitExperimentEvent / qualimaster::InitExperimentEvent
  + String pipeline(InitExperimentEvent)
  + Boolean ==(Any,Any)
  + Integer numWorkers(InitExperimentEvent)
  + Boolean !=(Any,Any)
  + String getPipeline(InitExperimentEvent)
  + Boolean <>(Any,Any)
  + Integer getNumExecutors(InitExperimentEvent)
  + Integer getNumWorkers(InitExperimentEvent)
  + Integer numExecutors(InitExperimentEvent)
* InitializationMode / qualimaster::InitializationMode
  + InitializationMode STATIC
  + InitializationMode DYNAMIC
  + InitializationMode ADAPTIVE
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* LoadShedderDescriptor / qualimaster::LoadShedderDescriptor
  + String getIdentifier(LoadShedderDescriptor)
  + Boolean ==(Any,Any)
  + String shortName(LoadShedderDescriptor)
  + String getShortName(LoadShedderDescriptor)
  + Boolean !=(Any,Any)
  + String identifier(LoadShedderDescriptor)
  + Boolean <>(Any,Any)
* LoadSheddingChangedMonitoringEvent / qualimaster::LoadSheddingChangedMonitoringEvent
  + String actualShedder(LoadSheddingChangedMonitoringEvent)
  + String getPipelineElement(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String getActualShedder(LoadSheddingChangedMonitoringEvent)
  + String getPipeline(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String quoteNull(String)
  + String unquoteNull(String)
  + Boolean <>(Any,Any)
  + String causeMessageId(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String pipeline(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + Boolean ==(Any,Any)
  + Any key(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String pipelineElement(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + Boolean !=(Any,Any)
  + String getShedder(LoadSheddingChangedMonitoringEvent)
  + Any getKey(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String shedder(LoadSheddingChangedMonitoringEvent)
  + String getCauseMessageId(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + new LoadSheddingChangedMonitoringEvent (String,String,String,String,String)
* LoadSheddingCommand / qualimaster::LoadSheddingCommand
  + String getPipelineElement(AbstractPipelineElementCommand)
  + Any getParameter(LoadSheddingCommand,LoadSheddingParameter)
  + Any parameter(LoadSheddingCommand,LoadSheddingParameter)
  + mapOf parameters(LoadSheddingCommand)
  + setIntParameter(LoadSheddingCommand,LoadSheddingParameter,Integer)
  + String getPipeline(AbstractPipelineCommand)
  + Boolean <>(Any,Any)
  + setIntParameter(LoadSheddingCommand,String,Integer)
  + String pipeline(AbstractPipelineCommand)
  + exec(CoordinationCommand)
  + Boolean ==(Any,Any)
  + String pipelineElement(AbstractPipelineElementCommand)
  + accept(LoadSheddingCommand,PseudoVoid)
  + Boolean !=(Any,Any)
  + String getShedder(LoadSheddingCommand)
  + setParameter(LoadSheddingCommand,String,Any)
  + String shedder(LoadSheddingCommand)
  + new LoadSheddingCommand (String,String,LoadShedderDescriptor)
  + new LoadSheddingCommand (String,String,String)
* LoadSheddingParameter / qualimaster::LoadSheddingParameter
  + Boolean ==(Any,Any)
  + String name(LoadSheddingParameter)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* MonitoringChangeCommand / qualimaster::MonitoringChangeCommand
  + String getPipelineElement(AbstractPipelineElementCommand)
  + String pipeline(AbstractPipelineCommand)
  + exec(CoordinationCommand)
  + mapOf getFrequencies(MonitoringChangeCommand)
  + Boolean ==(Any,Any)
  + String pipelineElement(AbstractPipelineElementCommand)
  + mapOf observables(MonitoringChangeCommand)
  + Boolean !=(Any,Any)
  + mapOf getObservables(MonitoringChangeCommand)
  + String getPipeline(AbstractPipelineCommand)
  + mapOf frequencies(MonitoringChangeCommand)
  + Boolean <>(Any,Any)
  + new MonitoringChangeCommand (String,String,mapOf,mapOf)
  + new MonitoringChangeCommand (String,String,mapOf)
  + new MonitoringChangeCommand (String,mapOf,mapOf)
  + new MonitoringChangeCommand (mapOf)
  + new MonitoringChangeCommand (mapOf,mapOf)
  + new MonitoringChangeCommand (String,mapOf)
* MonitoringFrequency / qualimaster::MonitoringFrequency
  + MonitoringFrequency CLUSTER\_MONITORING
  + MonitoringFrequency PIPELINE\_MONITORING
  + MonitoringFrequency PIPELINE\_NODE
  + MonitoringFrequency SOURCE\_AGGREGATION
  + MonitoringFrequency PIPELINE\_NODE\_RESOURCES
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* Observable / qualimaster::Observable
  + Boolean ==(Any,Any)
  + String name(Observable)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* ParallelismChangeCommand / qualimaster::ParallelismChangeCommand
  + String pipeline(AbstractPipelineCommand)
  + exec(CoordinationCommand)
  + Integer numberOfWorkers(ParallelismChangeCommand)
  + Boolean ==(Any,Any)
  + mapOf(String, ParallelismChangeRequest) incrementalChanges(ParallelismChangeCommand)
  + mapOf(String, Integer) getExecutors(ParallelismChangeCommand)
  + Boolean !=(Any,Any)
  + mapOf(String, Integer) executors(ParallelismChangeCommand)
  + Integer getNumberOfWorkers(ParallelismChangeCommand)
  + mapOf(String, ParallelismChangeRequest) getIncrementalChanges(ParallelismChangeCommand)
  + String getPipeline(AbstractPipelineCommand)
  + Boolean <>(Any,Any)
  + new ParallelismChangeCommand (String,Integer,mapOf(String, Integer))
  + new ParallelismChangeCommand (String,mapOf(String, ParallelismChangeRequest))
* ParallelismChangeRequest / qualimaster::ParallelismChangeRequest
  + Integer DELETE
  + Boolean ==(Any,Any)
  + String host(ParallelismChangeRequest)
  + Boolean !=(Any,Any)
  + Integer executorDiff(ParallelismChangeRequest)
  + Integer getExecutorDiff(ParallelismChangeRequest)
  + Boolean <>(Any,Any)
  + String getHost(ParallelismChangeRequest)
  + new ParallelismChangeRequest (Integer,String)
  + new ParallelismChangeRequest (Integer)
* ParameterChangeCommand / qualimaster::ParameterChangeCommand
  + String getPipelineElement(AbstractPipelineElementCommand)
  + String pipeline(AbstractPipelineCommand)
  + exec(CoordinationCommand)
  + String getParameter(ParameterChangeCommand)
  + Boolean ==(Any,Any)
  + String pipelineElement(AbstractPipelineElementCommand)
  + Boolean !=(Any,Any)
  + Any getValue(ParameterChangeCommand)
  + String parameter(ParameterChangeCommand)
  + String getPipeline(AbstractPipelineCommand)
  + Boolean <>(Any,Any)
  + Any value(ParameterChangeCommand)
  + new ParameterChangeCommand (String,String,String,Any)
* ParameterConfigurationAdaptationEvent / qualimaster::ParameterConfigurationAdaptationEvent
  + Boolean isUserTrigger(ParameterConfigurationAdaptationEvent)
  + String pipeline(ParameterConfigurationAdaptationEvent)
  + String getPipelineElement(ParameterConfigurationAdaptationEvent)
  + String getParameter(ParameterConfigurationAdaptationEvent)
  + Boolean ==(Any,Any)
  + String pipelineElement(ParameterConfigurationAdaptationEvent)
  + Boolean !=(Any,Any)
  + Any getValue(ParameterConfigurationAdaptationEvent)
  + String parameter(ParameterConfigurationAdaptationEvent)
  + String getPipeline(ParameterConfigurationAdaptationEvent)
  + Boolean <>(Any,Any)
  + Any value(ParameterConfigurationAdaptationEvent)
* PipelineCommand / qualimaster::PipelineCommand
  + PipelineOptions options(PipelineCommand)
  + String pipeline(AbstractPipelineCommand)
  + exec(CoordinationCommand)
  + Boolean ==(Any,Any)
  + PipelineOptions getOptions(PipelineCommand)
  + Boolean !=(Any,Any)
  + PipelineCommandStatus status(PipelineCommand)
  + PipelineCommandStatus getStatus(PipelineCommand)
  + String getPipeline(AbstractPipelineCommand)
  + Boolean <>(Any,Any)
  + new PipelineCommand (String,PipelineCommandStatus)
* PipelineOptions / qualimaster::PipelineOptions
* String SEPARATOR
* String KEY\_WORKERS
* String KEY\_MAINPIP
* String KEY\_PROFILINGMODE
* String PREFIX\_EXECUTOR
* String KEY\_WAIT\_TIME
* String SUFFIX\_PARLLELISM
* String SUFFIX\_TASKS
* String SUFFIX\_ARGUMENT
* String KEY\_ADAPTATION
* String PREFIX\_FREE
* taskParallelism(PipelineOptions,String,PseudoVoid)
* Integer getExecutorIntArgument(mapOf,String,String,Integer)
* mapOf toMap(PipelineOptions)
* Boolean isSubPipeline(PipelineOptions)
* Integer numberOfWorkers(PipelineOptions,Integer)
* Boolean getExecutorBooleanArgument(mapOf,String,String,Boolean)
* Boolean <>(Any,Any)
* Boolean isSubPipeline(String)
* Any getOption(PipelineOptions,String)
* Integer getExecutorParallelism(mapOf,String,Integer)
* Boolean executorBooleanArgument(mapOf,String,String,Boolean)
* Boolean !=(Any,Any)
* Type adaptationFilter(String)
* Boolean hasExecutorArgument(mapOf,String,String)
* Any option(PipelineOptions,String)
* String executorParallelismKey(String)
* String getExecutorStringArgument(mapOf,String,String,String)
* setExecutorArgument(PipelineOptions,String,String,Any)
* setExecutorArgument(PipelineOptions,String,Any)
* Integer getNumberOfWorkers(mapOf,Integer)
* setNumberOfWorkers(PipelineOptions,Integer)
* Integer getTaskParallelism(mapOf,String,Integer)
* executorParallelism(PipelineOptions,String,PseudoVoid)
* Boolean hasExecutorArgument(PipelineOptions,String,String)
* String getExecutorParallelismKey(String)
* Integer getWaitTime(PipelineOptions,Integer)
* setOption(PipelineOptions,String,Any)
* mapOf toConf(PipelineOptions,mapOf)
* merge(PipelineOptions,PipelineOptions)
* Integer getTaskParallelism(PipelineOptions,String,Integer)
* String mainPipeline(mapOf)
* String executorParamName(String)
* String adaptationFilterName(PipelineOptions)
* Any getExecutorArgument(PipelineOptions,String,String)
* String executorArgumentKey(String,String)
* executorLongArgument(mapOf,String,String,PseudoVoid)
* String getMainPipeline(PipelineOptions)
* Integer waitTime(PipelineOptions,Integer)
* Boolean isConfKey(String)
* Integer executorIntArgument(mapOf,String,String,Integer)
* setWaitTime(PipelineOptions,Integer)
* setExecutorParallelism(PipelineOptions,String,Integer)
* Boolean ==(Any,Any)
* Integer getNumberOfWorkers(PipelineOptions,Integer)
* setTaskParallelism(PipelineOptions,String,Integer)
* Boolean isInProfilingMode(PipelineOptions)
* String optionKey(String)
* String executorStringArgument(mapOf,String,String,String)
* String getMainPipeline(mapOf)
* getExecutorParallelism(PipelineOptions,String,PseudoVoid)
* Integer taskParallelism(PipelineOptions,String,Integer)
* Real executorDoubleArgument(mapOf,String,String,Real)
* Type adaptationFilter(PipelineOptions)
* String getOptionKey(String)
* Integer executorParallelism(mapOf,String,Integer)
* String getExecutorArgumentKey(String,String)
* Real getExecutorDoubleArgument(mapOf,String,String,Real)
* String mainPipeline(PipelineOptions)
* Integer numberOfWorkers(mapOf,Integer)
* Integer executorParallelism(PipelineOptions,String,Integer)
* getExecutorLongArgument(mapOf,String,String,PseudoVoid)
* String getAdaptationFilterName(PipelineOptions)
* Type getAdaptationFilter(PipelineOptions)
* getTaskParallelism(PipelineOptions,String,PseudoVoid)
* Integer taskParallelism(mapOf,String,Integer)
* markAsSubPipeline(PipelineOptions,String)
* toArgs(PipelineOptions,String)
* Integer getExecutorParallelism(PipelineOptions,String,Integer)
* Type getAdaptationFilter(String)
* enableProfilingMode(PipelineOptions)
* String getExecutorParamName(String)
* Any executorArgument(PipelineOptions,String,String)
* new PipelineOptions (PipelineOptions)
* new PipelineOptions (PseudoVoid)
* new PipelineOptions (Type)
* new PipelineOptions ()
* ProfileAlgorithmCommand / qualimaster::ProfileAlgorithmCommand
  + String getAlgorithm(ProfileAlgorithmCommand)
  + exec(CoordinationCommand)
  + String getFamily(ProfileAlgorithmCommand)
  + Boolean ==(Any,Any)
  + accept(ProfileAlgorithmCommand,PseudoVoid)
  + String algorithm(ProfileAlgorithmCommand)
  + Boolean !=(Any,Any)
  + String family(ProfileAlgorithmCommand)
  + Boolean <>(Any,Any)
  + new ProfileAlgorithmCommand (String,String)
* RegularAdaptationEvent / qualimaster::RegularAdaptationEvent
  + Boolean ==(Any,Any)
  + timestamp(RegularAdaptationEvent)
  + Boolean !=(Any,Any)
  + getTimestamp(RegularAdaptationEvent)
  + Boolean <>(Any,Any)
* ReplayAdaptationEvent / qualimaster::ReplayAdaptationEvent
  + String getPipelineElement(ReplayAdaptationEvent)
  + Real speed(ReplayAdaptationEvent)
  + String query(ReplayAdaptationEvent)
  + Real getSpeed(ReplayAdaptationEvent)
  + String startString(ReplayAdaptationEvent)
  + String getEndString(ReplayAdaptationEvent)
  + Boolean startReplay(ReplayAdaptationEvent)
  + String getQuery(ReplayAdaptationEvent)
  + String getPipeline(ReplayAdaptationEvent)
  + Boolean <>(Any,Any)
  + String getStartString(ReplayAdaptationEvent)
  + String pipeline(ReplayAdaptationEvent)
  + String endString(ReplayAdaptationEvent)
  + Boolean ==(Any,Any)
  + String pipelineElement(ReplayAdaptationEvent)
  + Integer ticket(ReplayAdaptationEvent)
  + Boolean !=(Any,Any)
  + Integer getTicket(ReplayAdaptationEvent)
  + Boolean getStartReplay(ReplayAdaptationEvent)
* ReplayChangedMonitoringEvent / qualimaster::ReplayChangedMonitoringEvent
  + String getPipelineElement(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + Boolean startReplay(ReplayChangedMonitoringEvent)
  + String getPipeline(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String quoteNull(String)
  + String unquoteNull(String)
  + Boolean <>(Any,Any)
  + String causeMessageId(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String pipeline(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + Boolean ==(Any,Any)
  + Any key(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + Integer ticket(ReplayChangedMonitoringEvent)
  + String pipelineElement(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + Boolean !=(Any,Any)
  + Integer getTicket(ReplayChangedMonitoringEvent)
  + Boolean getStartReplay(ReplayChangedMonitoringEvent)
  + Any getKey(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + String getCauseMessageId(AbstractPipelineElementEnactmentCompletedMonitoringEvent)
  + new ReplayChangedMonitoringEvent (String,String,Integer,Boolean,String)
* ReplayCommand / qualimaster::ReplayCommand
  + setReplayStartInfo(ReplayCommand,String,String,Real,String)
  + String getPipelineElement(AbstractPipelineElementCommand)
  + Real speedDbl(ReplayCommand)
  + String query(ReplayCommand)
  + Boolean startReplay(ReplayCommand)
  + String getQuery(ReplayCommand)
  + String getPipeline(AbstractPipelineCommand)
  + Boolean <>(Any,Any)
  + String pipeline(AbstractPipelineCommand)
  + exec(CoordinationCommand)
  + Real getSpeedDbl(ReplayCommand)
  + Boolean ==(Any,Any)
  + Integer ticket(ReplayCommand)
  + String pipelineElement(AbstractPipelineElementCommand)
  + accept(ReplayCommand,PseudoVoid)
  + Boolean !=(Any,Any)
  + Integer getTicket(ReplayCommand)
  + Boolean getStartReplay(ReplayCommand)
  + new ReplayCommand (String,String,Boolean,Integer)
* ResourceChangeAdaptationEvent / qualimaster::ResourceChangeAdaptationEvent
  + Boolean ==(Any,Any)
  + String resource(ResourceChangeAdaptationEvent)
  + Boolean !=(Any,Any)
  + ResourceChangeAdaptationEventStatus status(ResourceChangeAdaptationEvent)
  + ResourceChangeAdaptationEventStatus getStatus(ResourceChangeAdaptationEvent)
  + Boolean <>(Any,Any)
  + String getResource(ResourceChangeAdaptationEvent)
* ResourceChangedAdaptationEvent / qualimaster::ResourceChangedAdaptationEvent
  + Real oldValue(ResourceChangedAdaptationEvent,String)
  + Boolean ==(Any,Any)
  + String name(ResourceChangedAdaptationEvent)
  + String getName(ResourceChangedAdaptationEvent)
  + Real newValue(ResourceChangedAdaptationEvent,String)
  + Real getNewValue(ResourceChangedAdaptationEvent,String)
  + Boolean !=(Any,Any)
  + InfrastructurePart getPart(ResourceChangedAdaptationEvent)
  + Real getOldValue(ResourceChangedAdaptationEvent,String)
  + Boolean <>(Any,Any)
  + InfrastructurePart part(ResourceChangedAdaptationEvent)
* ResourceUsage / qualimaster::ResourceUsage
  + ResourceUsage USED\_MEMORY
  + ResourceUsage AVAILABLE\_MEMORY
  + ResourceUsage AVAILABLE\_FREQUENCY
  + ResourceUsage LOAD
  + ResourceUsage AVAILABLE\_MACHINES
  + ResourceUsage USED\_MACHINES
  + ResourceUsage AVAILABLE\_DFES
  + ResourceUsage USED\_DFES
  + ResourceUsage BANDWIDTH
  + ResourceUsage CAPACITY
  + ResourceUsage EXECUTORS
  + ResourceUsage TASKS
  + ResourceUsage HOSTS
  + ResourceUsage AVAILABLE\_CPUS
  + ResourceUsage USED\_CPUS
  + ResourceUsage AVAILABLE
  + Boolean ==(Any,Any)
  + String name(ResourceUsage)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* Scalability / qualimaster::Scalability
  + Scalability VOLUME
  + Scalability VELOCITY
  + Scalability VOLATILITY
  + Scalability VARIETY
  + Scalability ITEMS
  + Scalability PREDECESSOR\_ITEMS
  + Boolean ==(Any,Any)
  + String name(Scalability)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* ScheduleWavefrontAdaptationCommand / qualimaster::ScheduleWavefrontAdaptationCommand
  + String getPipelineElement(AbstractPipelineElementCommand)
  + String pipeline(AbstractPipelineCommand)
  + exec(CoordinationCommand)
  + Boolean ==(Any,Any)
  + String pipelineElement(AbstractPipelineElementCommand)
  + Boolean !=(Any,Any)
  + String getPipeline(AbstractPipelineCommand)
  + Boolean <>(Any,Any)
  + new ScheduleWavefrontAdaptationCommand (String,String)
* ShutdownAdaptationEvent / qualimaster::ShutdownAdaptationEvent
  + String pipeline(ShutdownAdaptationEvent)
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + String getPipeline(ShutdownAdaptationEvent)
  + Boolean isStopped(ShutdownAdaptationEvent)
  + Boolean <>(Any,Any)
* SourceVolumeAdaptationEvent / qualimaster::SourceVolumeAdaptationEvent
  + mapOf normalizedFindings(SourceVolumeAdaptationEvent)
  + String getSource(SourceVolumeAdaptationEvent)
  + Real getAverageDeviations(SourceVolumeAdaptationEvent)
  + Real allDeviations(SourceVolumeAdaptationEvent)
  + String getPipeline(SourceVolumeAdaptationEvent)
  + Real getAllDeviations(SourceVolumeAdaptationEvent)
  + String source(SourceVolumeAdaptationEvent)
  + Boolean <>(Any,Any)
  + mapOf getFindings(SourceVolumeAdaptationEvent)
  + String pipeline(SourceVolumeAdaptationEvent)
  + Boolean ==(Any,Any)
  + mapOf getNormalizedFindings(SourceVolumeAdaptationEvent)
  + Boolean !=(Any,Any)
  + mapOf findings(SourceVolumeAdaptationEvent)
  + Real averageDeviations(SourceVolumeAdaptationEvent)
* StartupAdaptationEvent / qualimaster::StartupAdaptationEvent
  + String getMainPipeline(StartupAdaptationEvent)
  + String pipeline(StartupAdaptationEvent)
  + Boolean ==(Any,Any)
  + Boolean !=(Any,Any)
  + String mainPipeline(StartupAdaptationEvent)
  + String getPipeline(StartupAdaptationEvent)
  + Boolean <>(Any,Any)
* TimeBehavior / qualimaster::TimeBehavior
  + TimeBehavior LATENCY
  + TimeBehavior THROUGHPUT\_ITEMS
  + TimeBehavior THROUGHPUT\_VOLUME
  + TimeBehavior ENACTMENT\_DELAY
  + Boolean ==(Any,Any)
  + String name(TimeBehavior)
  + Boolean !=(Any,Any)
  + Boolean <>(Any,Any)
* ViolatingClause / qualimaster::ViolatingClause
  + Real CLEARED
  + Real deviation(ViolatingClause)
  + String getOperation(ViolatingClause)
  + String variable(ViolatingClause)
  + Boolean isCleared(ViolatingClause)
  + String getVariable(ViolatingClause)
  + String getPipeline(ViolatingClause)
  + Boolean <>(Any,Any)
  + setPipeline(ViolatingClause,String)
  + Real getDeviationPercentage(ViolatingClause)
  + String pipeline(ViolatingClause)
  + Boolean ==(Any,Any)
  + String operation(ViolatingClause)
  + Boolean !=(Any,Any)
  + Real deviationPercentage(ViolatingClause)
  + Observable observable(ViolatingClause)
  + Real getDeviation(ViolatingClause)
  + Observable getObservable(ViolatingClause)

## Global operations

Global operations provide access to infrastructure operations such as source volume prediction, algorithm profiles, implementation information or resusable optimization / weighting functions.

* DecisionVariable obtainPipeline(Configuration,String)
* setOf(DecisionVariable) obtainPipeline(Configuration,Iterator)
* setOf(DecisionVariable) obtainPipeline(Configuration,Collection)
* DecisionVariable obtainPipeline(Configuration,String,Boolean)
* DecisionVariable obtainPipelineElement(Configuration,String)
* String obtainArtifactUrl(String,String,String)
* String obtainHardwareArtifactUrl(String)
* Boolean supportsTaskReallocation()
* Observable mapObservable(String)
* mapOf(String, Observable) mapObservable(setOf(String))
* sequenceOf(String) getSubTopologyComponents(String,String)
* mapOf(String, Integer) getSubTopologyScaling(String,String,Real,Boolean)
* mapOf(String, Integer) getSubTopologyScaling(String,String,Integer,Integer,Boolean)
* InitializationMode initializationMode()
* InitializationMode getInitializationMode()
* String implementationName(String,String)
* String getImplementationName(String,String)
* Boolean isSubPipeline(String,String)
* Real algorithmPrediction(String,String,String,Observable)
* mapOf(String, mapOf(Observable, Real)) algorithmPrediction(String,String,setOf(String),setOf(Observable),mapOf(Any, PseudoVoid))
* mapOf(String, mapOf(Observable, Real)) algorithmPrediction(String,String,setOf(String),setOf(Observable))
* Real algorithmPrediction(String,String,String,Observable,mapOf(Any, PseudoVoid))
* mapOf(String, mapOf(Observable, Real)) parameterPrediction(String,String,String,setOf(Observable),mapOf(Any, PseudoVoid))
* mapOf(String, Real) sourceVolumePrediction(String,String,sequenceOf(String))
* Real sourceVolumePrediction(String,String,String)
* ConstraintViolationAdaptationEvent toConstraintViolation(Real,Real)
* ConstraintViolationAdaptationEvent toConstraintViolation(SourceVolumeAdaptationEvent)
* mapOf(String, Real) weightAll(mapOf(String, mapOf(Observable, Real)),mapOf(Observable, Real))
* String weightingSelection(mapOf(String, mapOf(Observable, Real)),mapOf(Observable, Real))
* storeValueBinding(Configuration)
* storeValueBinding(Configuration,mapOf)