CCSDS MO M&C Service structures

# Action Service

## XTCE data structure comparison

ActionDefinitionDetails

This is a command definition.

## Operations

### OPERATION: updateDefinition

Only the description or the unit fields of the arguments shall be permitted to be changed.

## Data types

### Composite: ActionDefinitionDetails

The ActionDefinitionDetails structure holds the definition information of an action.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ActionDefinitionDetails | | |
| Field | Type | Nullable | Comment |
| name | MAL::Identifier | No | The name of the action. Must not be empty or the wildcard value. |
| description | MAL::String | No | The description of the action. |
| severity | Severity | No | Severity of the action. It is implementation specific what the meaning of the severity values are in a particular context. |
| progressStepCount | MAL::UShort | No | Total number of steps that will be reported if PROGRESS reporting is selected in the sent Action. 0 if PROGRESS reporting is not used. |
| arguments | List<ArgumentDefinitionDetails> | No | The list of argument definitions. |
| argumentIds | List<MAL::Identifier> | Yes | Optional list of argument definition identifiers. Allows the provider to verify that the correct arguments are being supplied when using the same field in the action instance. |

# Parameter Service

## XTCE data structure comparison

ParameterDefinitionDetails

This is a parameter definition.

## Operations

### OPERATION: enableGeneration

The provider should update the ParameterDefinition object in the COM archive if the generationEnabled field is changed.

### OPERATION: updateDefinition

Only the description, the raw unit, generationEnabled, or updateInterval fields shall be permitted to be changed.

## Data types

### Composite: ParameterDefinitionDetails

The ParameterDefinitionDetails structure holds a parameter definition. The conversion field defines the conditions where the relevant conversion is applied. For onboard parameters, the update interval should be a multiple of the minimum sampling interval of that parameter.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ParameterDefinitionDetails | | |
| Field | Type | Nullable | Comment |
| name | MAL::Identifier | No | The name of the parameter. Must not be empty or the wildcard value. |
| description | MAL::String | No | The description of the parameter. May be empty. |
| rawType | MAL::Octet | No | Holds the attribute short form part of the raw type of the parameter, e.g., for a MAL::String parameter it shall hold 15. |
| rawUnit | MAL::String | Yes | The unit for the raw value. If NULL then raw type has no unit. |
| generationEnabled | MAL::Boolean | No | Controls whether updates for this parameter are to be generated. |
| updateInterval | MAL::Duration | No | Periodic update interval. No periodic updates to be generated if this is set to '0'. |
| validityExpression | ParameterExpression | Yes | Expression that determines this parameter's validity state. Can be NULL if no validity check is required or validity is calculated by implementation-specific mechanisms. |
| conversion | ParameterConversion | Yes | If present then parameter has a converted type. |

### Composite: ParameterConversion

The ParameterConversion structure holds information about the conversions to be applied to a parameter.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ParameterConversion | | |
| Field | Type | Nullable | Comment |
| convertedType | MAL::Octet | No | Holds the attribute short form part of the converted type of the parameter, e.g., for a MAL::String parameter it shall hold 15. |
| convertedUnit | MAL::String | Yes | The converted parameter unit. If NULL then converted type has no unit. |
| conversionConditions | List<ConditionalReference> | No | The conversions to be applied. |

# Alert Service

## XTCE data structure comparison

AlertDefinitionDetails

This is an event definition.

## Operations

### OPERATION: enableGeneration

The provider should update the AlertDefinition object in the COM archive if the generationEnabled field is changed.

### OPERATION: updateDefinition

Only the description, the argument units, or the generationEnabled fields shall be permitted to be changed.

## Data types

### Composite: AlertDefinitionDetails

The AlertDefinitionDetails provides the definition of an alert including any argument definitions.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AlertDefinitionDetails | | |
| Field | Type | Nullable | Comment |
| name | MAL::Identifier | No | Alert name. Must not be empty or wildcard value. |
| description | MAL::String | No | The description of the alert. |
| severity | Severity | No | Severity of the alert. |
| generationEnabled | MAL::Boolean | No | Controls whether instances of this alert are to be generated. |
| arguments | List<ArgumentDefinitionDetails> | No | The list of argument definitions. |
| argumentIds | List<MAL::Identifier> | Yes | Optional list of argument definition identifiers. Allows the provider to verify that the correct arguments are being supplied. |

# Check Service

## XTCE data structure comparison

ConstantCheckDefinition

ReferenceCheckDefinition

DeltaCheckDefinition

LimitCheckDefinition

CompoundCheckDefinition

These are the limit (and other) parameter check definitions.

CheckLinkDetails

This holds the link between the parameter definition and the check definition.

## Operations

### OPERATION: enableCheck

The provider should update the CheckLink object in the COM archive if the checkEnabled field is changed.

### OPERATION: updateDefinition

Only the description, the maxReportingInterval, nominalCount, or violationCount fields shall be permitted to be changed in the CheckDefinition.

## Data types

### Composite: CheckDefinitionDetails

The CheckDefinitionDetails structure holds the definition of a check.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | CheckDefinitionDetails | | |
| Abstract | | | |
| Field | Type | Nullable | Comment |
| name | MAL::Identifier | No | The name of the check definition. Must not be empty or the wildcard value. |
| description | MAL::String | No | The description of the check. May be empty. |
| checkSeverity | Severity | No | Indicates the seriousness of the violation based on its possible negative consequences. |
| maxReportingInterval | MAL::Duration | No | Maximum interval that can elapse between generations of CheckResult updates. If this value expires, then a CheckResult is generated with the same state for the previous and current state. If set to '0', then no maximum reporting interval shall be applied. |
| nominalCount | MAL::UInteger | No | Number of consecutive valid samples passing the check for the check to be OK. |
| nominalTime | MAL::Duration | No | Duration that a parameter is continuously passing the check for the check to be OK. |
| violationCount | MAL::UInteger | No | Number of consecutive valid samples violating the check for the check to be in violation. |
| violationTime | MAL::Duration | No | Duration that a parameter is continuously violating the check for the check to be in violation. |

### Composite: ConstantCheckDefinition

The ConstantCheckDefinition structure holds the constant values to compare against for a consistency check.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ConstantCheckDefinition | | |
| Extends | CheckDefinitionDetails | | |
| Field | Type | Nullable | Comment |
| operator | COM::ExpressionOperator | No | The operator to be used to perform the check. |
| values | List<AttributeValue> | No | The set of constant values to be checked against. An empty list means that any value change triggers the check. |

### Composite: ReferenceCheckDefinition

The ReferenceCheckDefinition structure holds the key to another entity to compare against for a consistency check.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ReferenceCheckDefinition | | |
| Extends | CheckDefinitionDetails | | |
| Field | Type | Nullable | Comment |
| operator | COM::ExpressionOperator | No | The operator to be used to perform the check. |
| checkReference | ReferenceValue | No | The value to check against. |

### Composite: DeltaCheckDefinition

The DeltaCheckDefinition defines a delta transition check. It can be time-based or sample-based.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | DeltaCheckDefinition | | |
| Extends | CheckDefinitionDetails | | |
| Field | Type | Nullable | Comment |
| checkReference | ReferenceValue | No | The value to compare the current value against. |
| violateInRange | MAL::Boolean | No | If TRUE, then the safe (nonviolating) values lie outside the specified threshold range. |
| sampleBased | MAL::Boolean | No | If TRUE, then the delta values are calculated based on when the value updates; otherwise, the delta value is calculated based on an internally calculated time interval. |
| valueDelta | MAL::Boolean | No | If TRUE, then the thresholds contain value deltas. If FALSE, they contain percentage deltas. |
| lowerThreshold | MAL::Attribute | Yes | The lower threshold of the delta value. Must be of the correct type for the entity being checked. Must be a Float if percentage threshold. |
| upperThreshold | MAL::Attribute | Yes | The upper threshold of the delta value. Must be of the correct type for the entity being checked. Must be a Float if percentage threshold. |

### Composite: LimitCheckDefinition

The LimitCheckDefinition defines a high and low limit check. It is valid to supply only one limit; the other limit is assumed to be the relevant maximum supported by the type being checked in this case.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LimitCheckDefinition | | |
| Extends | CheckDefinitionDetails | | |
| Field | Type | Nullable | Comment |
| violateInRange | MAL::Boolean | No | If TRUE, then the safe (nonviolating) values lie outside the specified limits range. |
| lowerLimit | MAL::Attribute | Yes | The lower limit of the value. Must be of the correct type for the entity being checked. |
| upperLimit | MAL::Attribute | Yes | The upper limit of the value. Must be of the correct type for the entity being checked. |

### Composite: CompoundCheckDefinition

The CompoundCheckDefinition structure holds the object instance identifiers of one or more check link objects to monitor for a compound check.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | CompoundCheckDefinition | | |
| Extends | CheckDefinitionDetails | | |
| Field | Type | Nullable | Comment |
| minimumChecksInViolation | MAL::UInteger | No | The number of referenced checks that must be in violation for this check to be considered in violation. If set to '0' then all referenced checks must be in violation. |
| checkLinkIds | List<MAL::Long> | No | The set of check link objects that form the compound check. |

### Composite: CheckLinkDetails

The CheckLinkDetails structure represents the link from a check definition to a check result for a specific parameter.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | CheckLinkDetails | | |
| Field | Type | Nullable | Comment |
| checkEnabled | MAL::Boolean | No | TRUE if the check instance is enabled. |
| checkOnChange | MAL::Boolean | No | If TRUE then any change to state or value of the parameter, or the check condition will trigger a check evaluation. |
| useConverted | MAL::Boolean | No | If set to TRUE the converted value field of the parameter value should be used, otherwise the raw value field should be used. |
| checkInterval | MAL::Duration | No | The interval that a check should be applied. Only applicable if checkOnChange is FALSE. If '0', then no periodic checking shall be performed, and a check will be triggered by another mechanism. |
| condition | ParameterExpression | Yes | Should this check be applied, if NULL then always applied. |

### Composite: ReferenceValue

The ReferenceValue structure defines a value to compare against. A validCount of '1' and deltaTime of '0' would compare against the previous sample value.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ReferenceValue | | |
| Field | Type | Nullable | Comment |
| validCount | MAL::UShort | No | Number of valid samples that should be collected to update the reference value. |
| deltaTime | MAL::Duration | No | Delta time from now into the past from which the reference value should be sampled. |
| parameterId | COM::ObjectKey | Yes | The parameter to compare against. If NULL, then checked parameter should be compared against itself. |

# Statistic Service

## XTCE data structure comparison

StatisticFunctionDetails

This holds the description of the statistical function definition.

StatisticLinkDetails

This holds the link between the parameter definition and the statistical function definition.

## Operations

### OPERATION: enableGeneration

The provider should update the StatisticLink object in the COM archive if the generationEnabled field is changed.

### OPERATION: updateParameterEvaluation

All fields of the StatisticLinkDetails are updatable.

## Data types

### Composite: StatisticFunctionDetails

The StatisticFunctionDetails structure holds the details of the function.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | StatisticFunctionDetails | | |
| Field | Type | Nullable | Comment |
| name | MAL::Identifier | No | The name of the statistical function. |
| description | MAL::String | No | The description of the statistical function. |

### Composite: StatisticLinkDetails

The StatisticLinkDetails structure holds the sampling, reporting, and collection intervals for one parameter statistic function link.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | StatisticLinkDetails | | |
| Field | Type | Nullable | Comment |
| samplingInterval | MAL::Duration | No | The interval between samples of the parameter. |
| reportingInterval | MAL::Duration | No | The interval between for periodic updates being generated. If set to '0', then no periodic updates shall be sent. |
| collectionInterval | MAL::Duration | No | The collection and reset interval of the statistical evaluation for the linked parameter. If set to '0', then no periodic reset of the evaluation shall be performed. |
| resetEveryCollection | MAL::Boolean | No | If TRUE the evaluation will reset its value every collection interval. If FALSE it will maintain a moving evaluation of the function for the collection interval. |
| reportingEnabled | MAL::Boolean | No | TRUE if reporting of the evaluation instance is enabled. |

# Aggregation Service

## XTCE data structure comparison

AggregationDefinitionDetails

This holds the packet definition.

## Operations

### OPERATION: enableGeneration

The provider should update the AggregationDefinition object in the COM archive if the generationEnabled field is changed.

### OPERATION: enableFilter

The provider should update the AggregationDefinition object in the COM archive if the filterEnabled field is changed.

### OPERATION: updateDefinition

Only the description, generationEnabled, updateInterval, filterEnabled, or filterTimeout fields of the AggregationDefinition shall be permitted to be changed.

## Data types

### Composite: AggregationDefinitionDetails

The AggregationDefinitionDetails structure holds definition details of an aggregation.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AggregationDefinitionDetails | | |
| Field | Type | Nullable | Comment |
| name | MAL::Identifier | No | The name of the aggregation. Must not be empty or the wildcard value. |
| description | MAL::String | No | The description of the parameter. May be empty. |
| category | AggregationCategory | No | Category of the aggregation. |
| generationEnabled | MAL::Boolean | No | Controls whether updates for this aggregation are to be generated. |
| updateInterval | MAL::Duration | No | The interval between periodic updates on this aggregation. If this aggregation is not periodic, this field must be '0'. |
| filterEnabled | MAL::Boolean | No | Controls whether updates for this aggregation are to be filtered. |
| filteredTimeout | MAL::Duration | No | The maximum duration between filtered updates. If this value is exceeded, then an update is sent regardless of filtered thresholds. Ignored if not filtered. |
| parameterSets | List<AggregationParameterSet> | No | List containing the parameter sets which define the aggregation. |

### Composite: AggregationParameterSet

The AggregationParameterSet structure holds the identifier and optional filter for a parameter, or set of parameters, in an aggregation.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AggregationParameterSet | | |
| Field | Type | Nullable | Comment |
| domain | List<MAL::Identifier> | Yes | The domain of the parameters being referenced in this set of parameters, NULL if the same domain as the aggregation. |
| parameters | List<MAL::Long> | No | The list of object instance identifiers of the parameter definitions being included in the aggregation. |
| sampleInterval | MAL::Duration | No | The interval between samples of the parameters in the set. If '0' then just a single sample of the parameters is required per aggregation update. |
| periodicFilter | ThresholdFilter | Yes | If the AggregationParameterSet contains a single parameter then this field contains the filter to apply for filtered periodic updates when filters are applied. NULL if no filter required or this set contains more than one parameter. |

### Composite: ThresholdFilter

The ThresholdFilter structure holds the filter for a parameter.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ThresholdFilter | | |
| Field | Type | Nullable | Comment |
| thresholdType | ThresholdType | No | The type of filter to apply for filtered periodic updates when filters are applied. |
| thresholdValue | MAL::Attribute | No | Threshold value to apply. |

### ENUMERATION: AggregationCategory

AggregationCategory is an enumeration definition holding the categories of aggregations.

|  |  |  |
| --- | --- | --- |
| Name | AggregationCategory | |
| Enumeration Value | Numerical Value | Comment |
| GENERAL | 1 | General aggregation. |
| DIAGNOSTIC | 2 | Diagnostic aggregation. |

### ENUMERATION: ThresholdType

ThresholdType is an enumeration definition holding the types of filtering thresholds.

|  |  |  |
| --- | --- | --- |
| Name | ThresholdType | |
| Enumeration Value | Numerical Value | Comment |
| PERCENTAGE | 1 | Threshold value is a percentage. |
| DELTA | 2 | Threshold value is a delta. |

# Conversion Service

## XTCE data structure comparison

DiscreteConversionDetails

LineConversionDetails

PolyConversionDetails

RangeConversionDetails

These are the raw to engineered conversion definitions for parameters and command arguments.

## Operations

No updating of conversions should be possible.

## Data types

### Composite: DiscreteConversionDetails

The DiscreteConversionDetails structure holds a bidirectional conversion between raw and converted values. The first element of the pair is the raw value and the second is the converted value. Both sets of values must be unique.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | DiscreteConversionDetails | | |
| Field | Type | Nullable | Comment |
| mapping | List<MAL::Pair> | No | Defines a mapping between raw and converted values as a discrete set of points. The first entry in the pair is the raw value, and the second entry is the converted value. |

### Composite: LineConversionDetails

The LineConversionDetails structure is a bi-directional conversion between raw and converted values. It is defined by a series of points between which values are to be interpolated. The extrapolate attribute indicates if values can also be linearly extrapolated beyond the initial and final points.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LineConversionDetails | | |
| Field | Type | Nullable | Comment |
| extrapolate | MAL::Boolean | No | Indicates whether or not values can be extrapolated beyond the start and the end of the points. |
| points | List<MAL::Pair> | No | Defines the bi-directional conversion. The first attribute of the point is a raw value, and the second attribute is the converted value. |

### Composite: PolyConversionDetails

The PolyConversionDetails structure holds only forward (raw to converted) polynomial conversions. They are defined by a series of points for the polynomial coefficients.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PolyConversionDetails | | |
| Field | Type | Nullable | Comment |
| points | List<MAL::Pair> | No | The first attribute of a point is a MAL::Integer, being the degree of the polynomial; the second attribute is either a MAL::Float or a MAL::Double, being the coefficient of the term. |

### Composite: RangeConversionDetails

The RangeConversionDetails structure holds a range for a one-way conversion to convert between a continuous range to a discrete value. A range is defined as from this point up to, but not including, the next point.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | RangeConversionDetails | | |
| Field | Type | Nullable | Comment |
| points | List<MAL::Pair> | No | The first attribute in each pair is the raw range, and the second attribute is the converted value. |

# Group Service

## XTCE data structure comparison

GroupDetails

Holds references to other objects to simplify the specific operations such as enabling a set of Checks.

## Operations

Updating a Group object shall be limited to minor changes only, such as description.

More substantial changes require the delete and add operations.

## Data types

### Composite: GroupDetails

The GroupDetails structure holds the object type, domain, and set of object instance identifiers for a set of objects from another service.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | GroupDetails | | |
| Field | Type | Nullable | Comment |
| name | MAL::Identifier | No | Name of the group. |
| description | MAL::String | No | Description of the group. |
| objectType | COM::ObjectType | No | The object type of the objects referenced by this group. |
| domain | List<MAL::Identifier> | No | The domain of the objects being referenced by this group. |
| instanceIds | List<MAL::Long> | No | The list of object instance identifiers of the objects being referenced by this group. |

# Other Data types

### Composite: ArgumentDefinitionDetails

The ArgumentDefinitionDetails structure holds the details of an argument definition with a set of associated attributes, such as conversion used. The conversionCondition defines the conditions where the referenced conversion is applied.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ArgumentDefinitionDetails | | |
| Field | Type | Nullable | Comment |
| rawType | MAL::Octet | No | Holds the attribute short form part of the raw type of the argument, e.g., for a MAL::String argument it shall hold 15. |
| rawUnit | MAL::String | Yes | The unit for the raw value. |
| conversionCondition | List<ConditionalReference> | Yes | The conversion applied to the argument. |
| convertedType | MAL::Octet | Yes | Holds the attribute short form part of the converted type of the argument, e.g., for a MAL::String argument it shall hold 15. Must not be NULL if a conversion condition is supplied. |
| convertedUnit | MAL::String | Yes | The converted argument units. |

### Composite: ConditionalReference

The ConditionalReference structure holds a condition expression to be evaluated to determine if another item should be referenced. In the case that no test is required, i.e., the item should always be referenced, then the condition field should be set to NULL.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ConditionalReference | | |
| Field | Type | Nullable | Comment |
| condition | ParameterExpression | Yes | The expression indicates which entities are applicable for this check. If NULL, then the condition shall evaluate to TRUE. |
| referenceId | COM::ObjectId | No | Entity that should be referenced if the condition evaluates to TRUE or is NULL. |

### Composite: ParameterExpression

The ParameterExpression structure represents a simple expression between a parameter and a value for that parameter.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ParameterExpression | | |
| Field | Type | Nullable | Comment |
| parameterId | COM::ObjectKey | No | Identifies the parameter definition to be used in the expression. |
| operator | COM::ExpressionOperator | No | The expression operator. |
| useConverted | MAL::Boolean | No | If set to TRUE the converted value field of the parameter value should be used, otherwise the raw value field should be used. |
| value | MAL::Attribute | Yes | The value to be used in the expression. |

### Composite: AttributeValue

The AttributeValue structure holds an Attribute value. It allows a list of different Attribute types to be created whereas List<Attribute> would require the values to be all of the same type.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttributeValue | | |
| Field | Type | Nullable | Comment |
| value | MAL::Attribute | No | The argument value. Must not be NULL. NULL may be represented by having a NULL in place of the complete AttributeValue composite. |