INTERNATIONAL ELECTROTECHNICAL COMMISSION

\_\_\_\_\_\_\_\_\_\_\_\_

ASSET ADMINISTRATION SHELL FOR INDUSTRIAL APPLICATIONS

Part 5: Interfaces

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.

3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.

8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) IEC draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). IEC takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC had not received notice of (a) patent(s), which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at https://patents.iec.ch shall not be held responsible for identifying any or all such patent rights.

IEC 63278-5 Ed.1 has been prepared by IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

The text of this International Standard is based on the following documents:

|  |  |
| --- | --- |
| Draft | Report on voting |
| 65/XX/FDIS | 65/XX/RVD |

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement available at [www.iec.ch/members\_experts/refdocs](https://www.iec.ch/members_experts/refdocs). The main document types developed by IEC are described in greater detail at [www.iec.ch/publications](https://www.iec.ch/publications).

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under [webstore.iec.ch](https://webstore.iec.ch/?ref=menu) in the data related to the specific document. At this date, the document will be

* reconfirmed,
* withdrawn,
* replaced by a revised edition, or
* amended.

# AAS query language and AAS access rules

## General

Many use cases of the Asset Administration Shell involve a number of Asset Administration Shells at the same time. Executing the business logic on all potentially involved Asset Administration Shells solely by the AAS user application causes a considerable amount of transferred data objects and bandwidth. It is therefore useful to send parts of the selection conditions to the AAS administration interface(s) and AAS Submodel interface(s). The AAS query language enables AAS user applications to describe and handover their interests and computing environments to only respond with the specified data objects.

NOTE The specified query language shares concepts with the so-called Resource Query Language (RQL [[11]](https://industrialdigitaltwin.io/aas-specifications/IDTA-01002/v3.1.1/bibliography.html#bib11)). This language follows a simplified grammar and suitable expressiveness and is independent of any communication protocol. For the serialisation and de-serialisation, the same mechanisms are used as for all AAS interfaces and interface operations.

## Overview on AAS query language

Figure XXX shows the main elements of the AAS query language.

NOTE Some member of the classes of the AAS query language start with a dollar sign ($), following the conventions of RQL [11]:

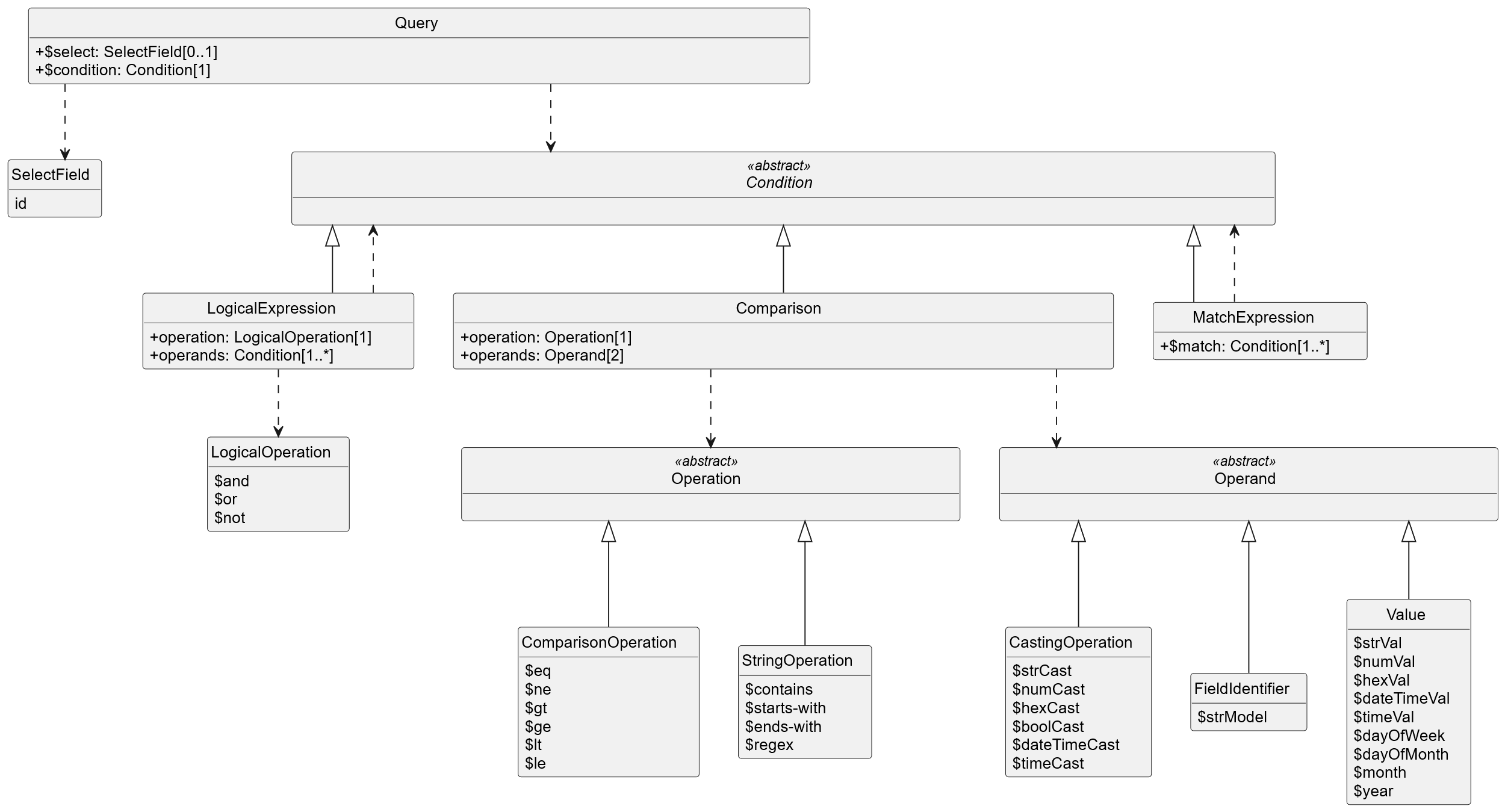


Figure 1 – Main elements of the AAS query language

A Query element is the root element of the AAS query language. In the optional member $select is set to "id", only Identifiers are returned by the query, elsewise entire AAS elements are serialized. The member $condition requires a composite Condition to be stated for filtering the entirety of all available AAS elements.

For the current edition of this standard, available AAS element types comprise AssetAdministrationShell, Submodels.

What is with SubmodelElements?

Each composite Condition may be composed of LogicalExpression elements, Comparison elements and MatchExpression elements.

LogicalExpression elements involve logical operators, such as "and", "or" and "not".

Comparison elements involve comparisons on ordered values such as "equal", "not equal", "greater", "greater or equal", "less than", "less than or equal" and string related comparisons such as "contains", "starts with", "ends with" and regular expressions.

Operands of these operations and comparisons comprise strings, numbers in different representations as well as boolean, time and date values.

The element of FieldIdentifier allows referring to specific AAS elements and members of these.

## Limitations

For the current edition of this standard, the following limitations apply:

1. The AAS Query Language focusses on Identifiables: It is only possible to formulate queries for Asset Administration Shells and Submodels.
2. Only selected attributes defined by the AAS data model (IEC 63278-2) can be used as conditions for returned objects: Usage of custom or vendor-specific attributes outside of the AAS specifications is generally not recommended and may lead to a rejections of the query by the receiving systems.
3. It is not possible to traverse through the specific fields of the SubmodelElements Entity (e.g. Entity/ statements), AnnotatedRelationshipElement (e.g. AnnotatedRelationshipElement/ annotations), or Operation (e.g. Operation/ outputVariables). Only the fields semanticId idShort, value, and valueType are available for SubmodelElements.

Search in hierarchy of AAS elements

As specified in IEC 63278-1, an AAS consists of a hierarchical structure. An AAS lists Submodels. A Submodel contains a list of SubmodelElements, which may include further SubmodelElements.

The query language allows to both address a specific instance of the data or to search all data in a computing environment. A specific instance is addressed by specifying the id of an Identifiable and the idShortPath to a Referable in a Submodel. If such explicit information is not specified, a hierarchical search is made. By the FieldIdentifier "$aas", conditions for all the relevant AAS are defined. By the FieldIdentifier "$sm", conditions for all the relevant Submodels are defined. FieldIdentifier "$sm" may be used with and without "$aas". If FieldIdentifier "$aas" and FieldIdentifier "$sm" are used together, only the Submodels referenced by the matching FieldIdentifier "$aas" are searched by the expression of the FieldIdentifier "$sm". The same search principle is used, when combining FieldIdentifier "$sm" and FieldIdentifier "$sme". In such case only the SubmodelElements which are part of matching Submodels by the expression of the FieldIdentifier "$sm" are searched by the expression of the FieldIdentifier "$sme". Several such hierarchical conditions may even be combined by a MatchExpression.

## Overview on AAS access rules

TBD

TBD

## Grammar for AAS query language and AAS access rules

The content and structure of the AAS Query Language is defined in the context-free Backus-Naur form (BNF, see Appendix XXX).

The text representation of the grammar is given by Table 1.

Table 1 – Text representation of the AAS query language and AAS access rules

<grammar> ::= <query> | <AllAccessPermissionRules>

<query> ::= <selectStatement>? <logicalExpression>

<selectStatement> ::= "$select" <ws> "id" <ws>

<logicalExpression> ::= <logicalNestedExpression> | <logicalOrExpression> | <logicalAndExpression> |

<logicalNotExpression> | <matchExpression> | <BoolLiteral> | <castToBool> | <singleComparison>

<logicalNestedExpression> ::= "(" <ws> <logicalExpression> ")" <ws>

<logicalOrExpression> ::= "$or" <ws> "(" <ws> <logicalExpression> ( "," <ws> <logicalExpression> )+ ")" <ws>

<logicalAndExpression> ::= "$and" <ws> "(" <ws> <logicalExpression> ( "," <ws> <logicalExpression> )+ ")" <ws>

<logicalNotExpression> ::= "$not" <ws> "(" <ws> <logicalExpression> ")" <ws>

<matchExpression> ::= ( "$match" <ws> "(" <ws> ( <singleComparison> | <matchExpression> ) ( "," <ws> ( <singleComparison> | <matchExpression> ) )\* ")" <ws> )

<singleComparison> ::=

<stringComparison> |

<numericalComparison> |

<hexComparison> |

<boolComparison> |

<dateTimeComparison> |

<timeComparison>

<allComparisons> ::= ( "$eq" | "$ne" | "$gt" | "$lt" | "$ge" | "$le" )

<stringComparison> ::=

( ( "$starts-with" | "$ends-with" | "$contains" | "$regex") <ws> "(" <ws> <stringOperand> <ws> "," <ws> <stringOperand> <ws> ")" <ws> ) |

( <stringOperand> <ws> <allComparisons> <ws> <stringOperand> <ws> ) |

( <stringOperand> <ws> <allComparisons> <ws> <FieldIdentifierString> <ws> ) |

( <FieldIdentifierString> <ws> <allComparisons> <ws> <stringOperand> <ws> )

<numericalComparison> ::=

( <numericalOperand> <ws> <allComparisons> <ws> <numericalOperand> <ws> ) |

( <numericalOperand> <ws> <allComparisons> <ws> <FieldIdentifierString> <ws> ) |

( <FieldIdentifierString> <ws> <allComparisons> <ws> <numericalOperand> <ws> )

<hexComparison> ::=

<hexOperand> <ws> <allComparisons> <ws> <hexOperand> <ws>

<boolComparison> ::=

<boolOperand> <ws> ( "$eq" | "$ne" ) <ws> <boolOperand> <ws>

<dateTimeComparison> ::=

<dateTimeOperand> <ws> <allComparisons> <ws> <dateTimeOperand> <ws>

<dateTimeToNum> ::=

( "$dayOfWeek" | "$dayOfMonth" | "$month" | "$year" ) <ws> "(" <ws> <dateTimeOperand> <ws> ")" <ws>

<timeComparison> ::=

<timeOperand> <ws> <allComparisons> <ws> <timeOperand> <ws>

<operand> ::= <stringOperand> | <numericalOperand> | <hexOperand> | <boolOperand> | <dateTimeOperand> | <timeOperand>

<stringOperand> ::=

<FieldIdentifierString> | <StringLiteral> | <castToString> | <SingleAttribute>

<numericalOperand> ::=

<NumericalLiteral> | <castToNumerical> | <dateTimeToNum>

<hexOperand> ::=

<HexLiteral> | <castToHex>

<boolOperand> ::=

<BoolLiteral> | <castToBool>

<dateTimeOperand> ::=

<DateTimeLiteral> | <castToDateTime> | <GlobalAttribute>

<timeOperand> ::=

<TimeLiteral> | <castToTime>

<castToString> ::=

"str" <ws> "(" <ws> <operand> <ws> ")" <ws>

<castToNumerical> ::=

"num" <ws> "(" <ws> <operand> <ws> ")" <ws>

<castToHex> ::=

"hex" <ws> "(" <ws> <operand> <ws> ")" <ws>

<castToBool> ::=

"bool" <ws> "(" <ws> <operand> <ws> ")" <ws>

<castToDateTime> ::=

"dateTime" <ws> "(" <ws> <stringOperand> <ws> ")" <ws>

<castToTime> ::=

"time" <ws> "(" <ws> ( <stringOperand> | <dateTimeOperand> ) <ws> ")" <ws>

<AllAccessPermissionRules> ::=

( "DEFATTRIBUTES" <ws> <StringLiteral> <ws> <AttributeGroup> <ws> )\*

( "DEFACLS" <ws> <StringLiteral> <ws> <ACL> <ws> )\*

( "DEFOBJECTS" <ws> <StringLiteral> <ws> <ObjectGroup> <ws> )\*

( "DEFFORMULAS" <ws> <StringLiteral> <ws> <Condition> <ws> )\*

( <AccessPermissionRule> <ws> )\*

<AccessPermissionRule> ::=

"ACCESSRULE:" <ws>

( <ACL> | <UseACL> ) <ws>

"OBJECTS:" <ws>

( <SingleObject> <ws> )\*

( <UseObjectGroup> <ws> )\*

"FORMULA:" <ws>

( <Condition> | <UseFormula> ) <ws>

( "FILTER:" <ws> <FragmentObject> <ws> ( <Condition> | <UseFormula> ) <ws> )?

<ACL> ::=

"ATTRIBUTES:" <ws>

( <SingleAttribute> <ws> )\*

( <UseAttributeGroup> <ws> )\*

"RIGHTS:" <ws> <Right> <ws> ( <Right> <ws> )\*

"ACCESS:" <ws> <Access> <ws>

<UseACL> ::=

"USEACLS" <ws> <StringLiteral> <ws>

<Right> ::=

"CREATE" | "READ" | "UPDATE" | "DELETE" | "EXECUTE" | "VIEW" | "ALL" | "TREE"

<Access> ::=

"ALLOW" | "DISABLED"

<SingleAttribute> ::=

<ClaimAttribute> | <GlobalAttribute> | <ReferenceAttribute>

<ClaimAttribute> ::=

"CLAIM" <ws> "(" <ws> <ClaimLiteral> <ws> ")"

<GlobalAttribute> ::=

"GLOBAL" <ws> "(" <ws> ( "LOCALNOW" | "UTCNOW" | "CLIENTNOW" | "ANONYMOUS" ) <ws> ")"

<ReferenceAttribute> ::=

"REFERENCE" <ws> "(" <ws> <ReferenceLiteral> <ws> ")"

<AttributeGroup> ::=

( <SingleAttribute> <ws> )\*

( <UseAttributeGroup> <ws> )\*

<UseAttributeGroup> ::=

"USEATTRIBUTES" <ws> <StringLiteral> <ws>

<SingleObject> ::=

<RouteObject> | <IdentifiableObject> | <ReferableObject> | <FragmentObject> | <DescriptorObject>

<RouteObject> ::=

"ROUTE" <ws> <RouteLiteral> <ws>

<IdentifiableObject> ::=

"IDENTIFIABLE" <ws> <IdentifiableLiteral> <ws>

<ReferableObject> ::=

"REFERABLE" <ws> <ReferableLiteral> <ws>

<FragmentObject> ::=

"FRAGMENT" <ws> <FragmentLiteral> <ws>

<DescriptorObject> ::=

"DESCRIPTOR" <ws> <DescriptorLiteral> <ws>

<ObjectGroup> ::=

( <SingleObject> <ws> )\*

| ( <UseObjectGroup> <ws> )\*

<UseObjectGroup> ::=

"USEOBJECTS" <ws> <StringLiteral> <ws>

<UseFormula> ::=

"USEFORMULAS" <ws> <StringLiteral> <ws>

<Condition> ::= <logicalExpression> <ws>

<DateTimeLiteral> ::= <datetime> <ws>

<TimeLiteral> ::= <time> <ws>

<datetime> ::= <date> <ws> ( "T" | " " ) <ws> <time> <ws> ( <timezone> <ws> )?

<date> ::= <year> <ws> "-" <ws> <month> <ws> "-" <ws> <day> <ws>

<year> ::= <digit> <ws> <digit> <ws> <digit> <ws> <digit> <ws>

<month> ::= <digit> <ws> <digit> <ws>

<day> ::= <digit> <ws> <digit> <ws>

<time> ::= <hour> <ws> ":" <ws> <minute> <ws> ( ":" <ws> <second> <ws> )? ( "." <ws> <fraction> <ws> )?

<timezone> ::= ( "Z" | ( "+" | "-" ) <ws> <hour> <ws> ":" <ws> <minute> <ws> )

<hour> ::= <digit> <ws> <digit> <ws>

<minute> ::= <digit> <ws> <digit> <ws>

<second> ::= <digit> <ws> <digit> <ws>

<fraction> ::= <digit>+ <ws>

<digit> ::= [0-9] <ws>

<StringLiteral> ::= "\"" ( [A-Z] | [a-z] | [0-9] | "/" | "\*" | "[" | "]" | "(" | ")" | " " | "\_" | "@" | "#" | "\\" | "+" | "-" | "." | "," | ":" | "$" | "^" | "\*" )+ "\""

<ClaimLiteral> ::= <StringLiteral>

<ReferenceLiteral> ::= <StringLiteral>

<RouteLiteral> ::= <StringLiteral>

<IdentifiableLiteral> ::= <StringLiteral>

<ReferableLiteral> ::= <StringLiteral>

<FragmentLiteral> ::= <StringLiteral>

<DescriptorLiteral> ::= <StringLiteral>

<NumericalLiteral> ::= ( "+" | "-" )? ( [0-9]+ ( "." [0-9]\* )? | "." [0-9]+ ) ( ( "e" | "E" )? [0-9]+ )

<HexLiteral> ::= "16#" ( [0-9] | [A-F] )+

<BoolLiteral> ::= "true" | "false"

<FieldIdentifier> ::= <FieldIdentifierString>

<FieldIdentifierString> ::= <FieldIdentifierAAS> | <FieldIdentifierSM> | <FieldIdentifierSME> | <FieldIdentifierCD> | <FieldIdentifierAasDescriptor> | <FieldIdentifierSmDescriptor>

<FieldIdentifierAAS> ::= "$aas#" ( "idShort" | "id" | "assetInformation.assetKind" | "assetInformation.assetType" | "assetInformation.globalAssetId" | "assetInformation." <SpecificAssetIdsClause> | "submodels." <ReferenceClause> )

<FieldIdentifierSM> ::= "$sm#" ( <SemanticIdClause> | "idShort" | "id" )

<FieldIdentifierSME> ::= "$sme" ( "." <idShortPath> )? "#" ( <SemanticIdClause> | "idShort" | "value" | "valueType" | "language" )

<FieldIdentifierCD> ::= "$cd#" ( "idShort" | "id" ) <ws>

<FieldIdentifierAasDescriptor> ::= "$aasdesc#" ( "idShort" | "id" | "assetKind" | "assetType" | "globalAssetId" | <SpecificAssetIdsClause> | "endpoints" ( "[" ( [0-9]\* ) "]" ) "." <EndpointClause> | "submodelDescriptors" ( "[" ( [0-9]\* ) "]" ) "." <SmDescriptorClause> )

<FieldIdentifierSmDescriptor> ::= "$smdesc#" <SmDescriptorClause>

<SmDescriptorClause> ::= ( <SemanticIdClause> | "idShort" | "id" | "endpoints" ( "[" ( [0-9]\* ) "]" ) "." <EndpointClause> )

<EndpointClause> ::= "interface" | "protocolinformation.href"

<ReferenceClause> ::= ( "type" | "keys" ( "[" ( [0-9]\* ) "]" ) ( ".type" | ".value" ) )

<SemanticIdClause> ::= ( "semanticId" | "semanticId." <ReferenceClause> )

<SpecificAssetIdsClause> ::= ( "specificAssetIds" ( "[" ( [0-9]\* ) "]" ) ( ".name" | ".value" | ".externalSubjectId" | ".externalSubjectId." <ReferenceClause> ) )

<idShortPath> ::= ( <idShort> ("[" ( [0-9]\* ) "]" )? ( "." <idShortPath> )\* )

<idShort> ::= ( ( [a-z] | [A-Z] ) ( [a-z] | [A-Z] | [0-9] | "\_" )\* )

<ws> ::= ( " " | "\t" | "\r" | "\n" )+

TEMPORARY BIBLIOGRAPHY

[11] "RQL: A resource query language for REST". Ariel Mashraki Version 1.3.0. March 2021. Online. Available: <https://github.com/a8m/rql>

[99] Peter Naur (editor): Revised Report on the Algorithmic Language ALGOL 60, 1963

# GENERATED CONTENT

Blank, to be added