

Event Query Language

Mandatory Statements' Syntax

Outline

- Component Definition Language
 - Environment Components
 - Sensor Network Components
 - Event Components
 - External Entities

Environment Syntax

- Platforms
 - Infrastructures
 - Location Maps
 - Locations and Inter-Location Relations
 - Devices
 - Hardware
 - Software
 - Services

CREATE Syntax (1/2)

```
CREATE PLATFORM ( [ ID ] <platform_id> [ , [ TYPE ] <type> = ‘infrastructure’ | ‘device’ ] ) ;
```

```
CREATE INFRASTRUCTURE ( [ ID ] <infrastructure_id> [ , [ LOCATION MAP ] <location_map_id> ] [ , { [ PLATFORM ] <platform_id> } ] ) ;
```

```
CREATE LOCATION MAP ( [ ID ] <location_map_id> [ , { [ LOCATION ] <location_id> } ] ) ;
```

```
CREATE LOCATION ( [ ID ] <location_id> [ , [ DESCRIPTION ] <description_id> ]
```

```
    [ , { ( [ RELATION ] <relation> , [ LOCATION ] <location_id> ) } ] ) ;
```

```
<relation> = ‘contains’ | ‘covers’ | ‘crosses’ | ‘equals’ | ‘includes’ | ‘isAbove’ | ‘isBelow’ | ‘isCloseTo’ | ‘isDisjointWith’ | ‘isFraFrom’ |  
‘isLeftOf’ | ‘isRightOf’ | ‘overlaps’ | ‘touches’ | <spatial_relation_id>
```

<spatial_relation_id> identifies an external spatial relationship defined in the application domain (external definition)

CREATE Syntax (2/2)

```
CREATE DEVICE ( [ ID ] <device_id> [ , { [ HARDWARE ] <hardware_id> } ] [ , { [ SOFTWARE ] <software_id> } ]  
[ , { [ SERVICE ] <service_id> } ] ) ;
```

```
CREATE HARDWARE ( [ ID ] <hardware_id> [ , [ DESCRIPTION ] <description_id> ] [ , { [ PROCESSOR ] <processor_id> } ]  
[ , { [ POWER SUPPLY ] <power_supply_id> } ] [ , { [ MEMORY ] <memory_id> } ]  
[ , { [ NETWORK INTERFACE ] <network_interface_id> } ] [ , { [ EXPANSION CARD ] <expansion_card_id> } ] ) ;
```

```
CREATE SOFTWARE ( [ ID ] <software_id> [ , [ DESCRIPTION ] <description_id> ] ) ;
```

```
CREATE SERVICE ( [ ID ] <service_id> [ , [ DESCRIPTION ] <description_id> ] ) ;
```

DESCRIPTION <description_id> is an external component that defines services, hardware, and software specifications

RENAME Syntax (1 / 1)

RENAME COMPONENT (<name> [TO] <new_name>) ;

COMPONENT = PLATFORM | INFRASTRUCTURE | LOCATION MAP | LOCATION |
DEVICE | HARDWARE | PROCESSOR | POWER SUPPLY | MEMORY | NETWORK INTERFACE | EXPANSION CARD |
SOFTWARE | SERVICE

DROP Syntax (1/1)

DROP COMPONENT (<name> [, CASCADE]) ;

COMPONENT = PLATFORM | INFRASTRUCTURE | LOCATION MAP | LOCATION |
DEVICE | HARDWARE | PROCESSOR | POWER SUPPLY | MEMORY | NETWORK INTERFACE | EXPANSION CARD |
SOFTWARE | SERVICE

ALTER ADD | REMOVE Syntax (1 / 2)

```
ALTER INFRASTRUCTURE ( [ ID ] <infrastructure_id> , ADD | REMOVE ( [ , [ LOCATION MAP ] <location_map_id> ]
[ , { [ PLATFORM ] <platform_id> } ] ) );
ALTER LOCATION MAP ( [ ID ] <location_map_id> , ADD | REMOVE { [ LOCATION ] <location_id> } ) ;
ALTER LOCATION ( [ ID ] <location_id> , ADD | REMOVE ( [ , [ DESCRIPTION ] <description_id> ]
[ , { ( [ RELATION ] <relation> , [ LOCATION ] <location_id> ) } ] ) );
ALTER DEVICE ( [ ID ] <device_id> , ADD | REMOVE ( [ , { [ HARDWARE ] <hardware_id> }
[ , { SOFTWARE <software_id> } ] [ , { SERVICE <service_id> } ] ) );
```

ALTER ADD | REMOVE Syntax (2/2)

```
ALTER HARDWARE ( [ ID ] <hardware_id> , ADD | REMOVE ( [ , [ DESCRIPTION ] <description_id> ]
[ , { [ PROCESSOR ] <processor_id> } ] [ , { [ POWER SUPPLY ] <power_supply_id> } ]
[ , { [ MEMORY ] <memory_id> } ] [ , { [ NETWORK INTERFACE ] <network_interface_id> } ]
[ , { [ EXPANSION CARD ] <expansion_card_id> } ] ) ) ;
ALTER SOFTWARE [ ID ] <software_id> , ADD | REMOVE [ DESCRIPTION ] <description_id> ;
ALTER SERVICE [ ID ] <service_id> , ADD | REMOVE [ DESCRIPTION ] <description_id> ;
```

ALTER MODIFY Syntax (1/2)

```
ALTER INFRASTRUCTURE ( [ ID ] <infrastructure_id> , MODIFY ( [ , [ LOCATION MAP ] [ <name> ] <location_map_id> ]
[ , { [ PLATFORM ] [ <name> ] <platform_id> } ] ) );
ALTER LOCATION MAP ( [ ID ] <location_map_id> , MODIFY { [ LOCATION ] [ <name> ] <location_id> } );
ALTER LOCATION ( [ ID ] <location_id> , MODIFY ( [ , [ DESCRIPTION ] [ <name> ] <description_id> ]
[ , { ( [ RELATION ] [ <name> ] <relation> , [ LOCATION ] [ <name> ] <location_id> ) } ] ) );
ALTER DEVICE ( [ ID ] <device_id> , MODIFY ( [ , { [ HARDWARE ] [ <name> ] <hardware_id> }
[ , { [ SOFTWARE ] [ <name> ] <software_id> ] } ] [ , { [ SERVICE ] [ <name> ] <service_id> } ] ) );
```

ALTER MODIFY Syntax (2/2)

```
ALTER HARDWARE ( [ ID ] <hardware_id> , MODIFY ( [ , [ DESCRIPTION ] [ <name> ] <description_id> ]
[ , { [ PROCESSOR ] [ <name> ] <processor_id> } ] [ , { [ POWER SUPPLY ] [ <name> ] <power_supply_id> } ]
[ , { [ MEMORY ] [ <name> ] <memory_id> } ]
[ , { [ NETWORK INTERFACE ] [ <name> ] <network_interface_id> } ]
[ , { [ EXPANSION CARD ] [ <name> ] <expansion_card_id> } ] ) ) ;
ALTER SOFTWARE ( [ ID ] <software_id> , MODIFY [ DESCRIPTION ] [ <name> ] <description_id> ) ;
ALTER SERVICE ( [ ID ] <service_id> , MODIFY [ DESCRIPTION ] [ <name> ] <description_id> ) ;
```

Sensor Network Syntax

- Properties & Observations
 - Observable Property
 - Observations
 - Scalar Observations
 - Textual
 - Multimedia Values
 - Objects
 - Metadata
- Sensors
 - Types
 - Mobile, Static
 - Locations
 - Coverage Areas
 - Properties
 - Observations

CREATE Syntax (1/2)

```
CREATE PROPERTY ( [ ID ] <property_id> [ , [ TYPE ] <type> = ' scalar ' | ' audio ' | ' image ' | ' video ' ]
                [ , { [ [ SCALAR | MEDIA ] OBSERVATION ] <observation_id> } ] ) ;
CREATE [ SCALAR | MEDIA ] OBSERVATION ( [ ID ] <observation_id>
                                         [ , [ DESCRIPTION ] <description_id> ]
                                         [ , ( [ DATA VALUE ] <data_value_id> | [ DATA OBJECT ] <data_object_id> , [ DATATYPE ] <datatype_id> ) ]
                                         [ , { ( [ METADATA TAG ] <metadata_tag> : [ METADATA VALUE ] <metadata_value> ) } ] )
                                         ) ;
```

CREATE Syntax (2/2)

```
CREATE SENSOR ( [ ID ] <sensor_id> [ , [ TYPE ] <type> = ' static ' | ' mobile ' ]  
  ( [ , WITH  
    [ , { [ DESCRIPTION ] <description_id> } ]  
    [ , [ LOCATION HISTORY ] <location_history> = { ( [ LOCATION ] <location_id> , [ TIME INTERVAL ] <ti> ) } ]  
    [ , [ COVERAGE HISTORY ] <coverage_history> = { ( [ COVERAGE AREA ] <coverage_area_id> , [ TIME INTERVAL ] <ti>  
  ) } ]  
  ] )  
  ( [ , SENSING { [ PROPERTY ] <property_id> } ] ) ( [ , PRODUCING { [ OBSERVATION ] <observation_id> } ] )  
  ( [ , HOSTED ON [ PLATFORM ] <platform_id> ] ) ) ;
```

RENAME Syntax (1 / 1)

RENAME COMPONENT (<name> [TO] <new_name>) ;

COMPONENT = PROPERTY | OBSERVATION | SENSOR

DROP Syntax (1/1)

DROP COMPONENT (<name> [, CASCADE]) ;

COMPONENT = PROPERTY | OBSERVATION | SENSOR

ALTER ADD | REMOVE Syntax (1/2)

```
ALTER PROPERTY ( [ ID ] <property_id> , ADD | REMOVE ( [ , [ TYPE ] <type> = ' scalar ' | ' audio ' | ' image ' | ' video ' ]  
[ , { [ [ SCALAR | MEDIA ] OBSERVATION ] <observation_id> } ] ) );
```

```
ALTER [ SCALAR | MEDIA ] OBSERVATION ( [ ID ] <observation_id> , ADD | REMOVE ( [ , [ DESCRIPTION ] <description_id> ]  
[ , ( [ DATA VALUE ] <data_value_id> | [ DATA OBJECT ] <data_object_id> , [ DATATYPE ] <datatype_id> ) ]  
[ , { ( [ METADATA TAG ] <metadata_tag> : [ METADATA VALUE ] <metadata_value> ) } ]  
) );
```

ALTER ADD | REMOVE Syntax (2/2)

```
CREATE SENSOR ( [ ID ] <sensor_id> , ADD | REMOVE ( [ , [ TYPE ] <type> = 'static' | 'mobile' ]  
          ( [ , WITH  
              [ , { [ DESCRIPTION ] <description_id> } ]  
              [ , [ LOCATION HISTORY ] <location_history> = { ( [ LOCATION ] <location_id> , [ TIME INTERVAL ] <ti> ) } ]  
              [ , [ COVERAGE HISTORY ] <coverage_history> = { ( [ COVERAGE AREA ] <coverage_area_id> , [ TIME INTERVAL ] <ti> ) } ]  
          ] )  
          ( [ , SENSING { [ PROPERTY ] <property_id> } ] ) ( [ , PRODUCING { [ OBSERVATION ] <observation_id> } ] )  
          ( [ , HOSTED ON [ PLATFORM ] <platform_id> ] )  
      ) );
```

ALTER MODIFY Syntax (1/2)

```
ALTER PROPERTY ( [ ID ] <property_id> , MODIFY ( [ , [ TYPE ] [ <name> ] <type> = ' scalar ' | ' audio ' | ' image ' | ' video ' ]
[ , { [ [ SCALAR | MEDIA ] OBSERVATION ] [ <name> ] <observation_id> } ] ) );
ALTER OBSERVATION ( [ ID ] <observation_id> , MODIFY (
[ , [ DESCRIPTION ] [ <name> ] <description_id> ]
[ , ( [ DATA VALUE ] [ <name> ] <data_value_id> | [ DATA OBJECT ] [ <name> ] <data_object_id> , [ DATATYPE ] [ <name> ] <datatype_id> )
[ , { ( [ METADATA TAG ] [ <name> ] <metadata_tag> : [ METADATA VALUE ] [ <name> ] <metadata_value> ) } ]
) );
```

ALTER MODIFY Syntax (2/2)

```
CREATE SENSOR ( [ ID ] <sensor_id> , MODIFY ( [ , [ TYPE ] [ <name> ] <type> = 'static' | 'mobile' ]
( [ , WITH
[ , { [ DESCRIPTION ] [ <name> ] <description_id> } ]
[ , [ LOCATION HISTORY ] [ <name> ] <location_history> = { ( [ LOCATION ] [ <name> ] <location_id> , [ TIME INTERVAL ] [ <name> ] <ti> ) }
[ , [ COVERAGE HISTORY ] [ <name> ] <coverage_history> = { ( [ COVERAGE AREA ] [ <name> ] <coverage_area_id> , [ TIME INTERVAL ] [ <name> ] <ti> ) }
] )
( [ , SENSING { [ PROPERTY ] [ <name> ] <property_id> } ] ) ( [ , PRODUCING { [ OBSERVATION ] [ <name> ] <observation_id> } ] )
( [ , HOSTED ON [ PLATFORM ] [ <name> ] <platform_id> ] )
) );
```

Event Syntax

- Event Definitions
 - Event Space
 - Event Features

CREATE Syntax (1/1)

```
CREATE EVENT ( [ ID ] <event_id> [ , [ EVENT SPACE ] <event_space_id> ] [ , USING { [ SENSOR ] <sensor_id> } ] ) ;  
CREATE EVENT SPACE ( [ ID ] <event_space_id> [ , { ( [ FEATURE ] <feature_id> [ , [ CONDITION ] <condition_id> ] ) } ]  
                    [ , { [ OBSERVATION ] <observation_id> } ] ) ;  
  
CREATE FEATURE ( [ ID ] <feature_id>  
                [ , [ DATATYPE ] <datatype> = ‘ integer ’ | ‘ float ’ | ‘ boolean ’ | ‘ date ’ | ‘ time ’ | ‘ date time ’ | ‘ character ’ | ‘ string ’ ]  
                [ , [ DISTANCE MEASURE ] <distance_measure_id> ]  
                [ , [ DEFAULT VALUE ] <value> ]  
                [ , [ DESCRIPTION ] <description_id> ] ) ;
```

CONDITION Syntax

```
CREATE CONDITION ( [ ID ] <condition_id> [ , { STATEMENT <statement_id> } ] ) ;
CREATE STATEMENT ( [ ID ] <statement_id> , ( [ OPERAND ] <operand_id>, [ OPERATOR ] <op> [ , [ OPERAND ] <operand_id> ] ) ) ;
CREATE OPERAND ( [ ID ] <operand_id> , ( [ TYPE ] <type> = ‘Temporal’ | ‘Spatial’ | ‘Other’ , [ VALUE ] <val> ) ) ;
<val> = <string> | [ LOCATION ] <location_id> | [ TIMESTAMP ] <ts> | [ TIME INTERVAL ] <ti> ;
<op> = [ COMPARISON ] <cop> | [ TEMPORAL ] <top> | [ SPATIAL ] <sop> | FUNCTION <function_id>
<cop> = ‘=’ | ‘<=’ | ‘>=’ | ‘<’ | ‘>’ | ‘not’ ;
<top> = ‘hasBeginning’ | ‘hasEnd’ | ‘inside’ | ‘intervalAfter’ | ‘intervalBefore’ | ‘intervalContains’ | ‘intervalDisjoint’ | ‘intervalDuring’ | ‘intervalEquals’ | ‘intervalFinishedBy’ |
‘intervalFinishes’ | ‘intervalIn’ | ‘intervalMeets’ | ‘intervalMetBy’ | ‘intervalOverlappedBy’ | ‘intervalOverlaps’ | ‘intervalStartedBy’ | ‘intervalStarts’ | [ TEMPORAL RELATION ]
<temporal_relation_id> ;
<sop> = ‘contains’ | ‘covers’ | ‘crosses’ | ‘equals’ | ‘includes’ | ‘isAbove’ | ‘isBelow’ | ‘isCloseTo’ | ‘isDisjointWith’ | ‘isFraFrom’ | ‘isLeftOf’ | ‘isRightOf’ | ‘overlaps’ | ‘
touches’ | [ SPATIAL RELATION ] <spatial_relation_id> ;
[ FUNCTION ] <function_id> ;
```

RENAME Syntax (1 / 1)

RENAME COMPONENT (<name> [TO] <new_name>) ;

COMPONENT = EVENT | EVENT SPACE | CONDITION | STATEMENT | OPERAND | FEATURE

DROP Syntax (1/1)

DROP COMPONENT (<name> [, CASCADE]) ;

COMPONENT = EVENT | EVENT SPACE | CONDITION | STATEMENT | OPERAND | FEATURE

ALTER ADD | REMOVE Syntax (1/1)

```
ALTER EVENT [ ID ] <event_id> , ADD | REMOVE ( [ , [ EVENT SPACE ] <event_space_id> ] [ , USING { [ SENSOR ] <sensor_id> } ] ) ;  
ALTER EVENT SPACE [ ID ] <event_space_id> , ADD | REMOVE ( [ , { ( [ FEATURE ] <feature_id> [ , [ CONDITION ] <condition_id> ] ) } ] [ , { [ OBSERVATION ] <observation_id> } ] ) ;  
ALTER FEATURE [ ID ] <feature_id> , ADD | REMOVE ( [ , [ DATATYPE ] <datatype> = 'integer' | 'float' | 'boolean' | 'date' | 'time' | 'date time' | 'character' | 'string' ]  
[ , [ DISTANCE MEASURE ] <distance_measure_id> ]  
[ , [ DEFAULT VALUE ] <value> ]  
[ , [ DESCRIPTION ] <description_id> ]  
) ;
```

ALTER MODIFY Syntax (1 / 1)

```
ALTER EVENT [ ID ] <event_id> , MODIFY ( [ , [ EVENT SPACE ] [ <name> ] <event_space_id> ] [ , USING { [ SENSOR ] [ <name> ] <sensor_id> } ] );
ALTER EVENT SPACE [ ID ] <event_space_id> , MODIFY ( [ , { ( [ FEATURE ] [ <name> ] <feature_id>
[ , [ CONDITION ] [ <name> ] <condition_id> ] ) } ] [ , { [ OBSERVATION ] [ <name> ] <observation_id> } ] );
ALTER FEATURE [ ID ] <feature_id> , MODIFY (
[ , [ DATATYPE ] [ <name> ] <datatype> = 'integer' | 'float' | 'boolean' | 'date' | 'time' | 'date time' | 'character' | 'string' ]
[ , [ DISTANCE FUNCTION ] [ <name> ] <distance_measure_id> ]
[ , [ DEFAULT VALUE ] [ <name> ] <value> ]
[ , [ DESCRIPTION ] [ <name> ] <description_id> ] );
```

Application Domain / External Entity Syntax

- External Entities
 - Concepts
 - Relations
 - Inter-concept relations for external entities (Concepts)
 - General relations for existing entities (Components)

CREATE | RENAME | DROP Syntax (1/1)

- CREATE

```
CREATE CONCEPT ( [ ID ] <concept_id> [ , { ELEMENT <element_id> } ] ) ;  
ELEMENT [ ID ] <element_id> = COMPONENT <component_id> | ATTRIBUTE ( <name>, <datatype> ) ;  
CREATE [ <name> ] RELATION ( [ ID ] <relation_id>  
    [ , { ( CONCEPT SOURCE <concept_id> , CONCEPT TARGET <concept_id> ) } ]  
    [ , { ( COMPONENT SOURCE <component_id> , COMPONENT TARGET <component_id> ) } ] ) ;
```

- RENAME

```
RENAME CONCEPT | RELATION ( <name> [ TO ] <new_name> ) ;
```

- DROP

```
DROP CONCEPT | RELATION ( <name> [ , CASCADE ] ) ;
```

ALTER ADD | REMOVE | MODIFY Syntax (1/1)

- ALTER ADD | REMOVE

```
ALTER CONCEPT [ ID ] <concept_id> , ADD | REMOVE ( [ , { ELEMENT <element_id> } ] ) ;
```

```
ALTER RELATION [ ID ] <relation_id> , ADD | REMOVE ( [ , { ( CONCEPT SOURCE <concept_id> , CONCEPT TARGET <concept_id> ) } ] [ , { ( COMPONENT SOURCE <component_id> , COMPONENT TARGET <component_id> ) } ] ) ;
```

- ALTER MODIFY

```
ALTER CONCEPT [ ID ] <concept_id> , MODIFY ( [ , { ELEMENT [ <name> ] <element_id> } ] ) ;
```

```
ALTER RELATION [ ID ] <relation_id> , MODIFY ( [ , { ( CONCEPT SOURCE [ <name> ] <concept_id> , CONCEPT TARGET [ <name> ] <concept_id> ) } ] [ , { ( COMPONENT SOURCE [ <name> ] <component_id> , COMPONENT TARGET [ <name> ] <component_id> ) } ] ) ;
```