

Toronto Water WMS Configuration Schema

Table of contents

Folder: 1-Schemas	3
Folder: A-entity_record_schema	3
00_common_definitions.yml	3
01_asset.yml	8
02_functional-location.yml	21
03_physical-location.yml	26
04_org-chart_group.yml	28
05_item_master.yml	30
06_tool_master.yml	35
07_service_item_master.yml	37
08_person.yml	37
09_qualification.yml	40
10_trade.yml	40
11_warranty.yml	42
12_service_contract.yml	43
32_job_plan.yml	45
33_PM.yml	51
34_FR_WR_WO.yml	56
36_work_order_documentation.yml	70
41_meter.yml	72
Folder: B-entity_class_object_schema	73
00_common_class_definitions.yml	73
01_asset_item_tool_class.yml	74
02_functional-location_class.yml	75
03_physical_location_class.yml	77
04_org-chart_group_class.yml	77
32_discrete_activity_class.yml	78
33_work_type_class.yml	78
Folder: 2-Classification_Trees	79
01_asset_classification.md	79
02_functional-location_classification.md	79
03_physical-location_classification.md	80
04_org-chart_group_classification.md	81
31_work_type.md	81
32_discrete_activity_classification.md	82
Folder: 3-System_Hierarchies	83
02_functional-location_hierarchy.md	83
03_physical-location_hierarchy.md	84
04_org-chart_group_hierarchy.md	84

Folder: 4-Class_Dependent_Specifications	85
Folder: A-asset_class_properties	85
01_pump.yml	85
02_motor.yml	87
03_valve.yml	90
04_breaker.yml	94
05_starter.yml	95
06_transformer.yml	96
07_HVAC.yml	97
08_blower_fan.yml	99
09_compressor.yml	100
10_generator.yml	100
11_UPS.yml	102
12_boiler.yml	103
13_pressure_vessel.yml	104
14_pressure_piping.yml	105
15_instrumentation.yml	107
Folder: B-role_class_properties	116
01_pump_role.yml	116
Folder: 5-Functions	116

Folder: 1-Schemas

Folder: A-entity_record_schema

00_common_definitions.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: common properties of all entities
4  $id:
5      ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/A-entity_record_schema/00_common_defin
6
7  definitions:
8
9  # ATTACHMENTS
10
11  attachment_def:
12    description:
13      oneOf:
14        - type: null
15        - type: array
16          items:
17            $ref: Maximo_DOCLINKS
18
19  # INVENTORY
20
21  rotating_property_def:
22    type: boolean
23    $comment: |
24      A rotating item is a trackable item, represented also as an asset in Maximo.
25      ↪ When the value of this field is true, we must serialize every instance of the
26      ↪ item. This commitment is beyond what we can presently achieve at TW. The more
27      ↪ pragmatic starting point considered in 2024 is as follows:
28        1) for the initial implementation, specify all items as non-rotating
29        2) an non-rotating item definition, containing the mfr, model, and
30      ↪ ordering_options, can be associated with any asset, through the the asset's
31      ↪ item_product_master_record property.
32        2) in the future, if we wish to convert the non-rotating item definition to
33      ↪ a rotating item definition, we would serialize all the assets associated to the
34      ↪ item definition, and convert them to rotating assets.
35  rule_spec:
36    - name: value of .properties."rotating item"
37      spec_ID: 4JKH1tw9gx
38      type: [validation, assertion]
39      specification: |
40        For a given item_x,
41        the value of item_x.properties."rotating flag" is set to false, for all
42      ↪ time
43      $comment:
44      status: specified
```

```
38 manufacturer_and_model_def:
39   type: object
40   properties:
41     manufacturer:
42       $ref: MaximoCompanyObject
43     model_and_sub-model:
44       oneOf:
45         - type: null
46         - type: string
47         description: For example, "Multilin 869"
48     product_version_or_model_year:
49       oneOf:
50         - type: null
51         - type: string
52         description: Identifies the specific version of the product model. For
53 ↪ example "v2" or "2023".
54     manufacturer_PN:
55       oneOf:
56         - type: null
57         - type: string
58         description: The manufacturer designator identifying the exact product item.
59 # FAILURE
60
61 failure_code:
62   type: object
63   properties:
64     code:
65       type: string
66     name:
67       type: string
68     description:
69       type: string
70     failure_code_type:
71       type: string
72     enum:
73       - problem
74       - cause
75       - remedy
76     site:
77       type: object
78       $ref: MaximoSiteObject
79     failure_classes:
80       oneOf:
81         - type: null
82         - type: array
83           items:
84             type: object
85             $ref: MaximoFailureClass
86     status:
87       type: string
88     enum:
89       - draft
```

```
90     - approved
91
92 # RESOURCE FOR PLAN AND WORK
93
94 item_requirement_definition:
95   properties:
96     item_reference:
97       $ref: "../05_item_master.yml"
98     required_quantity:
99       type: number
100     unit:
101       $ref: "../definitions/unit_of_measure"
102
103 tool_requirements_definition:
104   properties:
105     tool_reference: # reference for both stocked and un-stocked tool
106       $ref: "../06_tool_master.yml"
107     required_quantity:
108       type: number
109
110 service_requirement_definition:
111   properties:
112     service_reference:
113       $ref: "../07_service_item_master.yml"
114     required_quantity:
115       type: number
116     unit:
117       type: string
118       enum:
119         - hour
120         - instance
121
122 trade_requirement_definition:
123   properties:
124     trade_type:
125       $ref: "../B-entity_class_object_schema/08_trade_type.yml"
126     required_quantity:
127       type: number
128     qualification_requirement:
129       oneOf:
130         - type: null
131         - type: array
132       items:
133         $ref: "../B-entity_class_object_schema/09_qualification.yml"
134
135 # UNIT OF MEASURE
136
137 unit_of_measure:
138   description: Represents a unit of measure (UOM) used in inventory management to
139   ↪ track quantities of items.
140   properties:
141     name:
142       description: is the full name of the unit of measure.
```

```
142     type: string
143     $comment: e.g., "Each", "kilogram"
144   abbreviation:
145     description: is the unique identifier or code for the unit of measure.
146     type: string
147     $comment: e.g., "EA", "kg"
148
149 # FREQ DEFINITION
150
151 frequency_interval_definition:
152   properties:
153     frequency_quantity:
154       type: number
155     unit_of_time:
156       type: string
157       enum:
158         - minute
159         - hour
160         - day
161         - month
162         - year
163
164 # RECORD STATUS AND RETIREMENT
165
166 record_retirement_def:
167   properties:
168
169     record_is_active:
170       type: Boolean
171
172     retired_by:
173       oneOf:
174         - type: null
175         - $ref: "./08_person.yml"
176
177     reason_for_retirement:
178       read-only: true
179       #note: reason for requirement should be
180       oneOf:
181         - type: null
182         - type: string
183         enum:
184           - "record is a duplicate"
185           - "what record represents is gone"
186           - "what record represents is lost"
187           - "record added by mistake"
188           - "information out-dated"
189           - "information inaccurate"
190
191     date_of_actual_removal_or_departure:
192       description: is the date that the entity referent of the record was removed
193       ↪ or had departed
194       oneOf:
```

```
194     - type: null
195     - type: string
196       format: "date"
197
198   date_of_record_retirement:
199     description: is the date when the record was retired
200     oneOf:
201       - type: null
202       - type: string
203         format: "date"
204
205 # DUPLICATION HANDLING
206
207 duplicate_record_def:
208   oneOf:
209     - type: null
210     - type: array
211       items:
212         type: object
213         $comment: the object may other asset, functional-location,
↪ physical-location, ... records, depending on the context of where this property
↪ is used.
214
215 # METERS
216
217 meter_condition_definition:
218   properties:
219
220     meter:
221       description: is a selection of a pre-defined meter object.
222       $ref: "../41_meter.yml"
223
224     numeric_interval_value:
225       oneOf:
226         - type: null
227         - type: number
228
229     characteristic_trigger_value:
230       oneOf:
231         - type: null
232         - type: string
233
234 # COMPLIANCE REQUIREMENT DEFINITION
235
236 compliance_requirement:
237   properties:
238
239     name:
240       type: string
241       $comment: |
242         For example, 'ANSI Z358.1-2014 on weekly inspection of self-contained
↪ emergency wash equipment'.
```

```
244     requirement_detail:
245       description: provides all relevant descriptions
246       type: string
247       $comment: |
248         The following is an example of the requirement detail text for a
↪ compliance requirement.
249         applicable_asset_class:
250           - emergency eye-wash
251         requirement_source(s):
252           - ANSI Z358.1-2014 / Emergency Eyewash & Shower Standard / 4 Emergency
↪ Showers / 4.6 Maintenance and Training
253           ...
254           - ANSI Z358.1-2014 / Emergency Eyewash & Shower Standard / 4 Emergency
↪ Showers / 4.5 Installation
255           ...
256         source_content_guide:
257           - ANSI Z358.1-2014 4.6 states the requirement to check that shower
↪ still meets standards
258           - ANSI Z358.1-2014 4.5 states the standards to apply for ht check
259         perform_every:
260           - year
261
262     requirement_compliance_class:
263       description: indicates the level of compliance, with legislative being the
↪ top
264       $ref: "#/compliance_class"
265
266     compliance_class:
267       type: string
268       enum:
269         - legislative
270         - corporate policy
```

01_asset.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: asset
4  $id:
↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/01_as
5  type: object
6
7  $comment: >
8
9  properties:
10
11    ID:
12      type: string
13      description: A read-only UUID, generated by the system, to uniquely identify the
↪ asset record.
14
```



```
15  name:
16      type: string
17      description: The human readable short description of the asset.
18      $comment: |
19          Assumption: an non-is a specific commercial product is always built on site
↵  for a specific purpose, and would permanently occupy a functional-location. An
↵  example is an aeration tank.
20
21  class:
22      $ref: "../B-entity_class_object_schema/01_asset_item_tool_class.yml"
23      description: indicates the class to which this asset is an instance.
24
25  GIS_object_ID:
26      type: string
27      integration: GIS
28      description: is unique ID of the GIS asset record, which represents the same
↵  asset as this record.
29
30  specification_data:
31      type: object
32      description: is a set of specification property data. (The applicable properties
↵  are dependent on the definition made in the chosen class.)
33
34  inferred_classes:
35      oneOf:
36          - type: null
37          - type: array
38          items:
39              $ref: "../B-entity_class_object_schema/01_asset_item_tool_class.yml"
40      read-only: true
41      integration: data-hub
42      description: indicates the complex classes to which this asset is an instance. A
↵  complex class is defined with reference to a primitive class plus other
↵  attributes. An example of a complex class is the TSSA high-pressure boiler
↵  class, which is made with reference to th primitive class boiler.
43
44  # STATE AND STATUS GROUP OF PROPERTIES
45
46  physical_status:
47      type: string
48      description: Indicates whether the asset is present at the City, and more
↵  precisely, at its working location. It also indicates when the knowledge of the
↵  asset's presence is missing (i.e., it is missing or lost).
49      enum:
50          - planned
51          - in possession
52          - installed
53          - abandoned in place
54          - removed from possession
55          - missing
56          - lost
57      $comment: |
58          This data field is not nullable because the lack of knowledge is explicitly
↵  expressed as "missing" or "lost", and the non-existence is expressed as
↵  "planned" or "removed from possession".
```

```
59
60   operating_state:
61     type: string
62     description: Indicates whether the asset is available for doing the work that it
↪ is assigned at a given moment. Only applies to asset that is assigned to a
↪ functional-location, user-group, or user.
63     enum:
64       - available (up)
65       - unavailable (down)
66       - not applicable
67     $comment: |
68       To data architect and implementer: the "not applicable" value is important,
↪ because when we are reporting on equipment uptime, we need to know about the
↪ periods in which the operating state is not applicable. For example, if the
↪ asset is not assigned to any functional-location, org-chart_group, or anyone.
69
70 # OWNER, OPERATOR, MAINTAINER GROUP
71
72   owned_by_the_group:
73     oneOf:
74       - type: null
75       - $ref: "./04_org-chart_group.yml"
76     description: Denotes the org-chart_group that owns the asset.
77     integration: GIS
78
79   owned_by_the_unlisted_group:
80     oneOf:
81       - type: null
82       - type: string
83     description: name of an org-chart group that is not found in the current list
↪ (and should be added)
84     integration: GIS
85     $comment: |
86       To WIM, this data field should be added to the GIS to allow an asset record to
↪ be enter into the system, even if the org-chart_group that owns it had not been
↪ added to the value list.
87
88   maintenance_group:
89     oneOf:
90       - type: null
91       - $ref: "./04_org-chart_group.yml"
92     description: group responsible for the overall maintenance of the asset - for
↪ example, a unit, work area, or crew.
93
94   operator_group:
95     oneOf:
96       - type: null
97       - $ref: "./04_org-chart_group.yml"
98     description: group responsible for the operation of the asset
99
100 # ASSIGNMENTS
101
102   assignment_type:
```

```
103     oneOf:
104       - type: string
105       - type: null
106     description:
107     enum:
108       - to a functional-location
109       - to a user group
110       - to a single user
111       - not assigned
112     $comment: |
113       This property is added to assist with the interpretation of the null value in
114       ↪ the "assigned_to_asset_functional-location", "assigned_to_tool_user_group", or
115       ↪ "assigned_to_tool_user_group" property. If the value here is "not assigned",
116       ↪ then we know the asset is not assigned to anything. If the value here is null,
117       ↪ we do not know whether this asset is assigned to anything.
118
119     assigned_to_asset_functional-location:
120
121       oneOf:
122         - $ref: "../02_functional-location.yml"
123         - type: null
124       description: functional-location that the asset is designated to play. This
125       ↪ value persists even if the asset is temporarily removed from the location of the
126       ↪ functional-location (for reasons such as repair).
127
128     assigned_to_tool_user_group:
129       oneOf:
130         - $ref: "../04_org-chart_group.yml"
131         - type: null
132       description: A group of people, such as a facility, work area, or crew to whom
133       ↪ the asset is assigned for use. Indicates the assignment of an asset (usually a
134       ↪ tool) that does not have a system functional-location.
135
136     assigned_to_tool_user:
137       oneOf:
138         - $ref: "../08_person.yml"
139         - type: null
140       description: Indicates the assignment of an asset (usually a tool) that does not
141       ↪ have a system functional-location.
142
143 # LOCATION
144
145     installation_or_parking_location:
146       oneOf:
147         - $ref: "../03_space.yml"
148         - type: null
149
150     service_address_or_coordinate:
151       oneOf:
152         - $ref: MaximoServiceAddressObject
153         - type: null
154       $comment: |
155         this is referencing Maximo's native service address object
```

```
147
148   parent_asset:
149     oneOf:
150       - $ref: "./01_asset.yml"
151       - type: null
152     description: >
153       Indicates the larger discrete asset or defined collection of assets, to which
↪ this asset is a part of. NOTE: this property is not meant to be used for
↪ specifying the system hierarchy parent. That property is found on the
↪ functional-location record, not the asset record.
154     $comment: |
155       This field is commonly used when the asset is a part of a skid, structural
↪ tank, or switchgear cabinet, in which the asset parent in the system hierarchy
↪ should be the line entity. As such we will using this field to track that the
↪ asset is also a part of a physical assembly. We would also be using this field
↪ to capture a serialized rotating component as a part of another discrete asset.
↪ This field can also be used to indicate an asset membership in a Defined
↪ Collection of Assets.
156
157 # PRODUCT AND TOOL ASSOCIATION GROUP
158
159   is_a_commercially_available_product:
160     type: boolean
161     description: An asset is made under as a product of a commercial entity, as
↪ opposed to an asset that is assembled on site.
162     $comment: No null value allowed because this information is self-evident
163
164   is_a_tool:
165     type: boolean
166     description: a tool is enables or enhances the ability of a human agent to
↪ perform a piece of maintenance, repair, testing, and investigative work. "true"
↪ value would designate the asset as a rotating tool, which allows the asset to be
↪ 1. reserved for work, or 2. assigned to a staff or group (which includes fixed
↪ tools).
167     $comment: Note that this property was changed from "mobile" because this
↪ designation also applies to fixed tools, such as machine shop or lab tools. All
↪ of these assets fall within the definition of a tool.
168
169   is_mobile:
170     oneOf:
171       - type: boolean
172       - type: null
173
174   item_product_master_record:
175     oneOf:
176       - $ref: "./05_item_master.yml"
177       - type: null
178     description: This field links the asset to an item record that defines a
↪ specific commercial product. By effect, it also deems to asset to be a rotating
↪ item.
179
180   tool_product_master_record:
181     oneOf:
```

```
182     - $ref: "./06_tool_master.yml"
183     - type: null
184     description: A association with a master record designates the asset as a
↪ stocked tool, which allows the tool to be checked into a storeroom and tracked
↪ as a part of an inventory. Without an association, the tool would be
↪ non-stocked.
185
186 # MANUFACTURER AND MODEL
187
188     commercial_product_information:
189         oneOf:
190             - type: null
191             -
↪ $ref: "./00_common_definitions.yml#/definitions/plain-text_manufacturer_and_model_def"
192
193 # DATE PROPERTIES
194
195     construction_contract_number:
196         oneOf:
197             - type: string
198             - type: null
199     description: The construction_contract_number (usually RFQ#) assigned by the
↪ City
200
201     first_day_of_City_operation:
202         oneOf:
203             - type: string
204             - type: null
205     description: The day that the asset is turned over to the City from a
↪ contractor, or if the City installed the asset itself - the day the asset enters
↪ operation after testing is completed.
206     $comment: |
207         This usually coincides with "warranty start date". However, if the asset is
↪ not delivered through a project, "warranty start date" may be empty.
208
209     OEM_serial:
210         oneOf:
211             - type: string
212             - type: null
213     description: The serial number, affixed on the asset, designated by the
↪ manufacturer.
214     $comment: |
215         ASMP Discussion Log: The serial number is only populated when an asset
↪ experiences a movement (except for movement for removal), or when it is being
↪ check into a storeroom. Therefore, when the value of the OEM_serial is null, it
↪ represents the fact that we do not know what the serial number is (and whether
↪ it has a serial number at all).
216
217     purchase_cost_in_CAD:
218     description: the original purchase cost of the asset (not necessary if the asset
↪ is associated with a item master record)
219     oneOf:
220         - type: number
```

```
221     - type: null
222
223   asset_photos:
224     oneOf:
225       - type: array
226         items:
227           oneOf:
228             - type: null
229             - type: array
230               items:
231                 $ref: "./00_common_definitions.yml/attachment_def"
232       - type: null
233
234   # COMMON RECORD PROPERTIES
235
236   duplicate_record_of:
237     $ref: "./00_common_definitions.yml#/definitions/duplicate_record_def"
238
239   record_retirement_information:
240     $ref: "./00_common_definitions.yml#/definitions/record_retirement_def"
241
242   # BACKGROUND PROPERTIES POPULATED AUTOMATICALLY
243
244   TW_asset_group:
245     oneOf:
246       - type: string
247       - type: null
248     invisible: true
249     read-only: true
250     enum:
251       - Drinking Water Network
252       - Drinking Water Treatment Plants
253       - Waste and Storm Water Network
254       - Wastewater Treatment Plants
255       - Independent Building
256       - Multiple Major Systems
257     $comment: |
258       Note that this property is populated automatically, and not available for user
↵ to edit. Use-case: asset from the GIS will not be indexed on the hierarchy. The
↵ main use of this property is to provide a simple handle term, when one needs to
↵ summarize the collection of all assets imported from a certain layer(s) the GIS.
259
260   #####
261   # RULES
262   #####
263
264   rule_spec:
265
266     - name: Vertical Asset ID
267       spec_ID: 41JeoQuvex
268       involves_prop: [ID]
269       type: [assertion]
270       specification: |
```

```
271     Upon the creation of a new vertical facility asset record generate a unique ID
↪ (such as UUID Ver4)
272     checked_on: 2024-08-15
273     $comment: |
274         UUID has a distinct advantage over a simple serial number - we do not need a
↪ script to check for repetition. For instance, when onboarding assets from
↪ another system or a spreadsheet, we don't need to check the WMS to see if the ID
↪ was already taken.
275
276 - name: Asset Naming
277   spec_ID: 4ykh0m_Dle
278   involves_prop: [name]
279   type: [assertion]
280   specification: |
281       if asset_x.properties."is_a_commercially_available_product" = TRUE
282           asset_x.properties.name is the semi-colon delimited concatenation of:
283           - asset_x.properties.class.properties."class name"
284           - asset_x.properties."item_product_master_record".properties.product
↪ manufacturer company
285           -
↪ asset_x.properties."item_product_master_record".properties.model_and_sub-model
286           - asset_x.properties."item_product_master_record".properties.product
↪ configuration code
287           - asset_x.properties."OEM_serial"
288       elif asset_x.properties."is_a_commercially_available_product" = FALSE
289           asset_x.properties.name is the semi-colon delimited concatenation of:
290           - asset_x.properties.class.properties."class name"
291           -
↪ asset_x.properties."assigned_to_asset_functional-location".properties.name
292       # NOTE: actual script should contain additional condition handle formatting of
↪ the name text when there is missing data in any concatenated property.
293   status:
294       checked: 2024-08-15
295
296 - name: Exclusion Of Part And Material Classes From Asset Classification
297   involves_prop: [class]
298   spec_ID: V15NNHZuxl
299   type: [validation, UI]
300   specification: |
301       Assertion Part:
302       For all assets "asset_x",
303       the value of (asset_x.class.properties.only_used_as_a_part asset) must be
↪ FALSE
304       UI Part:
305       In all asset classification search or selection screens, eliminate or filter
↪ out all classes "class_y",
306       where (class_y.properties.only_used_as_a_part asset) is TRUE
307   checked_on: 2024-08-15
308
309 - name: Valid Values of owned_by_the_group Property
310   involves_prop: [owned_by_the_group]
311   spec_ID: 410N2dr_xx
312   type: [validation, UI]
```

```
313     specification: |
314         - The valid range of values for selection includes the first or second of the
↪ org-chart group hierarchy, specified in the
↪ (/TWDM/3-System_Hierarchies/04_org_hierarchy.md) . For examples,
315         - first level example: York Region,
316         - second level example: Toronto Water
317         - The UI must only present the valid range of values to the users for
↪ selection, and the valid range of values must be presented as a hierarchy.
318     checked_on: 2024-08-19
319
320 - name: Inheriting The Asset's Maintenance And Operator Group Values From Its
↪ functional-location
321     involves_prop: [operator_group, maintenance_group]
322     spec_ID: VJ1QRgIclg
323     specification: |
324         - if the value of asset_x.properties.assigned_to_asset_functional-location is
↪ functional-location_y, then
325             inherit the value of
326                 - asset_x.properties.maintenance_group
327                 - asset_x.properties.operator_group
328             from the same properties of functional-location_y
329     checked_on: 2024-08-20
330
331 - name: Rendering of assignment_type Data Field.
332     involves_prop: [assignment_type]
333     spec_ID: 4yARRuvOex
334     type: [UI]
335     description: |
336         the options of this property should be presented as radial button
337     status: specified
338
339 - name: Valid Assignment of an Asset
340     spec_ID: NyrzGKwuel
341     type: [validation, assertion, UI]
342     description: |
343         If asset_x.properties."assignment_type" = "to a functional-location", then
344         - asset_x.properties."assigned_to_asset_functional-location" must NOT =
↪ null;
345         - asset_x.properties."assigned_to_tool_user" must = null
346         - asset_x.properties."assigned_to_tool_user_group" must = null
347     elif .properties."assignment_type" = "to a user group", then
348         - asset_x.properties."assigned_to_tool_user_group" must NOT = null;
349         - asset_x.properties."assigned_to_asset_functional-location" must = null
350         - asset_x.properties."assigned_to_tool_user" must = null
351     elif asset_x.properties."assignment_type" = "to a single user", then
352         - asset_x.properties."assigned_to_tool_user" must NOT = null;
353         - asset_x.properties."assigned_to_asset_functional-location" must = null
354         - asset_x.properties."assigned_to_tool_user_group" must = null
355     elif asset_x.properties."assignment_type" = null, then
356         - asset_x.properties."assigned_to_asset_functional-location" must = null
357         - asset_x.properties."assigned_to_tool_user" must = null
358         - asset_x.properties."assigned_to_tool_user_group" must = null
359     Also, in the UI screen, disable the properties that should = null
```



```
360     status: specified
361
362 - name: asset present at site must have location information on record
363   involves_prop: [service_address_or_coordinate]
364   spec_ID: 01J5R2F9ARJDM3RMGE9WYZWVFE
365   type: [validation]
366   specification: |
367     if the value of asset_x.properties.physical_status is either
368       - "in possession", or
369       - "installed"
370     then at least one of the following properties must NOT be null
371       - asset_x.properties.installation_or_parking_location
372       - asset_x.properties.service_address_or_coordinate
373   check_on: 2024-08-20
374
375 - name: Automatic Value Assignment to
↪ properties."is_a_commercially_available_product"
376   involves_prop: [is_a_commercially_available_product]
377   spec_ID: 4Jg2gYS0ee
378   type: [assertion]
379   specification: |
380     - Upon record creation, set the value to TRUE.
381     - Upon a asset_x.properties.class value change or a re-run of the Maximo rule
↪ processor,
382       if asset_x.properties.class.properties."non-manufactured" = TRUE;
383       set the value to TRUE;
384       else set the value to FALSE.
385   status: [specified]
386
387 - name: Default Value of is_a_tool
388   involves_prop: [is_a_tool]
389   spec_ID: 41sz7KSdxe
390   type: [assertion]
391   specification: |
392     - Upon record creation, set the default value to FALSE.
393     - Upon a asset_x.properties.class value change or a re-run of the Maximo rule
↪ processor,
394       if asset_x.properties.class.properties.tool = TRUE;
395       then set the value to TRUE;
396   status: [specified, checked]
397
398 # - name: If an asset is commercially available but not a tool, then it must have
↪ mfr and model information.
399 #   spec_ID: VJY43yI9lx
400 #   involves_prop: item_product_master_record
401 #   type: [assertion, UI]
402 #   specification: |
403 #     if asset_x.properties.is_a_commercially_available_product = TRUE AND
↪ asset_x.properties.is_a_tool = FALSE, then
404 #       - (asset_x.properties."item_product_master_record") is NOT null
405 #       - enable (asset_x.properties."item_product_master_record") in UI
406 #     else
407 #       - (asset_x.properties."item_product_master_record") is null
```

```
408 # - disable (asset_x.properties."item_product_master_record") in UI
409 # status: TBS
410
411 - name: valid item master record in .properties.item_product_master_record
412   spec_ID: VJGKn1I9ex
413   involves_prop: [item_product_master_record]
414   type: [validation]
415   specification: |
416     For asset_x.properties.item_product_master_record,
417     only accept a master record whose value of
↵ asset_x.properties.generic_or_specific_product is "specific commercial product".
↵
418   status: [specified, checked]
419   $comment: related to 4y3dRfLcee
420
421 - name: Serial on Mobile Assets
422   involves_prop: [is_mobile]
423   spec_ID: EyA3sYa9le
424   type: [validation]
425   specification: |
426     For any asset_x,
427     if the value of asset_x.properties.is_a_tool is TRUE, and the value of
↵ asset_x.properties.is_mobile is also TRUE, then
428     the value of asset_x.properties.OEM_serial cannot be null.
429   check_on: 2024-08-20
430
431 - name: when to enable the tool_product_master_record
432   involves_prop: [tool_product_master_record]
433   spec_ID: NyQBbeL9xl
434   specification: |
435     if asset_x.properties."is_a_tool" = TRUE
436     then enable (asset_x.properties."tool_product_master_record") property.
↵
437   status: [specified, checked]
438 - name: valid value of asset_x.properties.tool_product_master_record
439   spec_ID: NyFFWlUc1l
440   type: [validation]
441   specification: |
442     only accept a tool_product_master_record whose
↵ .properties.generic_or_specific_product property value is "specific commercial
↵ product"
443   checked_on: 2024-08-20
444
445 - name: an asset may either be associated with a tool or an asset, not both
446   involves_prop: [tool_product_master_record]
447   spec_ID:
448   type:
449   specification:
450   status: TBS
451
452 - name: Linear Asset Id
453   type: [assertion]
454   spec_ID: Vku-67dDxx
```

```
455   involves_prop: [ID]
456   specification: |
457     Upon the creation of a new asset record corresponding to a record in TWAG,
↪   through the Maximo-TWAG integration,
458     populate the TWAG_asset record's "Facility ID" value into the "ID".
459   checked_on: 2024-08-15
460   $comment: see comment for rule 41JeoQuvex.
461
462   - name: Asset must have a start of operation date info before we can indicate that
↪   it is operationally available.
463     spec_ID: NyG2nzL5xg
464     type: [validation]
465     specification: |
466       if both of the following properties are null
467         - (asset_x.properties."first date of City operation")
468         - (asset_x.properties."warranty start date")
469       then the value of (asset_x.properties."operating_state") CANNOT be
↪   "available (up)"
470
471   - name: asset can be assigned exclusively to either a functional-location, user,
↪   or user group
472     spec_ID: EkD-ZmIceg
473     type: [validation]
474     specification: |
475       only one of the following properties can have value (i.e., not null) at any
↪   given time. (It is also okay for all of them to be null)
476       - asset_x.properties."assigned_to_asset_functional-location"
477       - asset_x.properties."assigned_to_tool_user"
478       - asset_x.properties."assigned_to_tool_user_group"
479
480
481   - name: Consistency Between Operating State And Assignment Values
482     spec_ID: 410Fxr8ceg
483     type: [validation, assertion]
484     specification: |
485       if an asset does not have a value in any of the following properties (i.e.,
↪   all nulls),
486       - asset_x.properties."assigned_to_asset_functional-location"
487       - asset_x.properties."assigned_to_tool_user"
488       - asset_x.properties."assigned_to_tool_user_group"
489       then the value of (asset_x.properties."operating_state") must be "not
↪   assigned work". The opposite must also be true.
490     errorMessage: "An asset NOT assigned to a functional-location, user, or user
↪   group should not be operating and therefore would not have an operating_state"
491
492   - name: Consistency Between Asset's Physical Status, Operating State, And
↪   Assignments
493     spec_ID: NyG2nzL5xg
494     type: [assertion, validation]
495     specification: |
496       If the value of (asset_x.properties."physical_status") is NEITHER of the
↪   following
497       - "installed"
```

```

498     - "in possession"
499     then the following properties would take on the stated values
500     asset_x.properties."operating_state" = "not applicable"
501     asset_x.properties."assigned_to_asset_functional-location" = null
502     asset_x.properties."assigned_to_tool_user" = null
503     asset_x.properties."assigned_to_tool_user_group" = null
504
505 - name: Asset Can Only Be Assigned To A Discrete Asset functional-location
506   spec_ID: 4yBXuH8qle
507   type: [validation]
508   specification: |
509     if (asset_x.properties."assigned_to_asset_functional-location") is NOT null
510     then
511 ↪ (asset_x.properties."assigned_to_asset_functional-location".properties.class.properties."d
512 ↪ asset functional-location") = TRUE
513   errorMessage: an asset can only be assigned to a discrete asset
514 ↪ functional-location
515
516 - name: inheriting the asset location information from its functional-location
517   spec_ID: NJdGTHLqeg
518   type: [assertion]
519   specification: |
520     For an asset, asset_x, if
521     all of the following are true:
522     - asset_x.properties."operating_state" = "installed"
523     - asset_x.properties."assigned_to_asset_functional-location" is NOT null
524     and one of the following is true
525     -
526 ↪ asset_x.properties."assigned_to_asset_functional-location".properties."asset_installation_
527 ↪ is NOT null
528     -
529 ↪ asset_x.properties."assigned_to_asset_functional-location".properties."service_address_or_
530 ↪ is NOT null
531     then
532     (asset_x.properties."installation_or_parking_location") would be set to
533 ↪ the value of
534     (asset_x.properties."assigned_to_asset_functional-location".properties."asset_installation
535     (asset_x.properties."service_address_or_coordinate") would be set to the
536 ↪ value of
537     (asset_x.properties."assigned_to_asset_functional-location".properties."service_address_or
538
539 - name: over-write of manufacturer and model information
540   spec_ID: 01J5RPPEKJCP11NBRW3A4XCKF7
541   specification: |
542     if the value of either
543     status: TBS
544
545 - name: a commercially available asset must be associated with manufacturer and
546 ↪ model information
547   spec_ID: 4y3dRfLcee
548   type: [validation]

```

```
539     exempt_grandfather: true
540     specification: |
541         if the value of asset_x.properties.is_a_commercially_available_product is TRUE
542             then NONE of the following properties can be null
543                 - asset_x.properties.item_product_master_record
544                 - asset_x.properties.manufacturer_name
545                 - asset_x.properties.product_model_information
546
```

02_functional-location.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: functional-location
4  $id:
5  ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/02_as
6  type: object
7  properties:
8
9      ID:
10     type: string
11     description: is the unique ID of the functional-location known as the "tag
12     ↪ number" or "entity number" in Avantis's vocabulary. (Avantis is the a WMS).
13
14     name:
15     type: string
16     description: is a structured description of the functional-location.
17
18     parent:
19     $ref: "./02_functional-location.yml"
20     description: references the functional-location that is served by the larger
21     ↪ asset, which physically subsumes the asset serving this functional-location.
22
23     GIS_object_ID:
24     type: string
25     integration: GIS
26     description: is unique ID of the GIS record representing the same pumping
27     ↪ station, chamber, or ... as this *functional-location* record in Maximo.
28
29     class:
30     $ref: "B-entity_class_object_schema/02_functional-location_classification.md"
31     description: denotes the broad types of asset that may play the
32     ↪ functional-location (e.g., "breaker functional-location"). It also denotes the
33     ↪ useful function provided by an asset in the functional-location to the larger
34     ↪ system (e.g., the "tie-breaker functional-location" provides tie-breaking
35     ↪ function).
36
37     inferred_classes:
38     oneOf:
39     - type: null
```

```
33     - type: array
34       items:
35         $ref:
↪ "B-entity_class_object_schema/02_functional-location_classification.md"
36       items:
37         type: string
38       read-only: true
39       description: indicates the complex classes to which this asset is an instance.
40
41   specification_data:
42     type: object
43     $ref: "."
44     description: is a set of functional performance specification data. (The
↪ applicable specifications are dependent on the definition made in the chosen
↪ class.)
45
46   functional-location_status:
47     type: string
48     enum:
49       - specified
50       - active
51       - eliminated
52     description: is the life-cycle status of a functional-location.
53     $comment: |
54       "specified" means the functional-location is conceived and exists in some
↪ specification or design documentation; "active" means the necessary supports
↪ exist for an asset to serve in the functional-location and function of the asset
↪ being utilized; "eliminated" represents a negation of either or both conditions
↪ of the active status.
55
56   installation_location:
57     oneOf:
58       - $ref: "./03_space.yml"
59       - type: null
60     description: refers to the physical-location in which the asset serving the
↪ functional-location would be installed.
61
62   service_relations:
63     description: identifies the asset, by the functional-location it is occupying,
↪ that this functional-location is serving. For example, given a motor starter
↪ functional-location, the value in this data field identifies the
↪ functional-location of the motor controlled by that motor starter.
64     oneOf:
65       - type: null
66       - type: object
67         properties:
68
69           serving:
70             $ref: "./02_functional-location.yml"
71
72           service_type:
73             oneOf:
74               - type: null
```

```
75     - type: string
76     enum:
77       - "powers"
78       - "supplies material"
79       - "supplies energy"
80       - "controls or regulates"
81     $comment: the list is not exhaustive.
82
83   service_address_or_coordinate:
84     oneOf:
85       - $ref: MaximoServiceAddressObject
86       - type: null
87     description: is the geo-coordinate or the nearest street address of the asset.
88
89 # OWNER, OPERATOR, MAINTAINER GROUP
90
91   operator_group:
92     oneOf:
93       - type: null
94       - $ref: "./04_org-chart_group.yml"
95     description: group responsible for the operation of the asset in the
96 ↪ functional-location.
97
98   maintenance_group:
99     oneOf:
100       - type: null
101       - $ref: "./04_org-chart_group.yml"
102     description: group responsible for the overall maintenance of the asset - for
103 ↪ example, a unit, work area, or crew.
104
105 #note:
106 # on: the absence of of a owned_by_the_group property for functional-locations
107 # content: assets that are occupying a functional-location are presumed to be
108 ↪ owned by the City.
109
110   inherit_operator_group_value:
111     type: boolean
112     default_value: true
113
114   inherit_operator_group_from_parent:
115     type: boolean
116     default_value: true
117
118   inherit_maintenance_group_from_parent:
119     type: boolean
120     default_value: true
121
122   operational_criticality:
123     oneOf:
124       - $ref: "#/definitions/criticalityRatingDef"
125       - type: null
126     description: A functional-location bears high operational criticality if the
127 ↪ loss of the asset in the functional-location will either reduce throughput or
128 ↪ product quality (but not product safety) of the larger system.
```

```
124
125   protective_function_criticality:
126     oneOf:
127       - $ref: "#/definitions/criticalityRatingDef"
128       - type: null
129     description: A functional-location bears protective function criticality if the
↪ loss of one of its protective functions (i.e., regulatory/control/protection or
↪ containment function) will either result in a consequential release of hazard or
↪ the loss of a capability to mitigate a greater level hazard.
130
131 # COMMON RECORD PROPERTIES
132
133 duplicate_record_of:
134   $ref: "./00_common_definitions.yml#/definitions/duplicate_record_def"
135
136 record_retirement_information:
137   $ref: "./00_common_definitions.yml#/definitions/record_retirement_def"
138
139
140 #####
141 # LOCAL OBJECT DEFINITION
142 #####
143
144 definitions:
145   criticality rating definition:
146     type: object
147     properties:
148       rating:
149         type: integer
150       description:
151         type: string
152     enum:
153       - rating: 1
154         description: TBD
155       - rating: 2
156         description: TBD
157       - rating: 3
158         description: TBD
159       - rating: 4
160         description: TBD
161       - rating: 5
162         description: TBD
163
164
165 #####
166 # RULES
167 #####
168
169 rule_spec:
170
171   - name: ID of Linear Assets Represented as functional-location in Maximo
172     spec_ID: Vku-67dDxx
173     involves_prop: [ID]
```



```
174   type: [triggered action]
175   specification:
176     trigger: replication creation of assetY record from the TWAG
177     action: apply Facility ID from TWAG as ID
178   status: false
179
180 - name: ID of Linear Assets Represented as functional-location in Maximo
181   spec_ID: Vku-67dDxx
182   involves_prop: [ID]
183   type: [triggered action]
184   specification:
185     - if:
186       oneOf:
187         - assetfunctional-locationClass:
188           properties:
189             className:
190               const: pumping station
191         - assetfunctional-locationClass:
192           properties:
193             className:
194               const: water treatment plant
195         - assetfunctional-locationClass:
196           properties:
197             className:
198               const: large chamber
199       then:
200         required: GIS_object_ID
201   status: false
202
203   #[]RULE VkiDyJcSxg: Before a functional-location can be eliminated, all
↪ children, as well as the descendants of the functional-location in the hierarchy
↪ must also be eliminated. $comment: a procedure should to be created to allow the
↪ recursive elimination of a functional-location and all of its children.
204
205   #[]RULE VygDCOFrx1: When a functional-location is "eliminated", it must no
↪ longer be visible in any view of the functional-location hierarchy. (Its entire
↪ branch must not be available either, because all of its descendants would be
↪ eliminated as well.)
206   #[]Review with To ASMP: with this rule, we no longer need to have a hierarchy
↪ branched for retired functional-locations.
207
208   #[]RULE EkP5qy5Sx1: If change auditing cannot be turned on, then when a
↪ functional-location record status is "eliminated", all the specification in the
↪ record's data fields must be frozen.
209
210   #[]RULE:
211   # IN COMMON LANGUAGE: At any given time, each functional-location may only have
↪ a single asset assigned to it (i.e., associated with the functional-location via
↪ the asset's "assigned_to_asset_functional-location" property).
```

03_physical-location.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: physical-location
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/03_sp
6  type: object
7  properties:
8
9    ID:
10
11    type: string
12    description: An unique ID
13    $comment: In the future, this value should be validated with a regular
14    ↪ expression.
15
16    parent:
17
18    $ref: "./03_space.yml"
19    description: The larger physical-location that completely contains this
20    ↪ physical-location.
21
22    name:
23
24    type: string
25    description: Short name. Should be one that is commonly use by staff in
26    ↪ communication. For example, "boardroom"
27
28    complete_name:
29
30    type: string
31    read-only: true
32    rule_spec:
33      - spec_ID:
34        name: complete name generation
35        id: 01JFVNSOYDFD7K5DP4NVMSKTY8
36        status:
37        $comment: |
38          Automatically generated by the system and not editable. The value is name to
39          ↪ that of its parent, its grandparent ... all the way up that facility
40          ↪ physical-location.
41
42    enclosed_by_asset:
43
44    oneOf: [$ref: "./01_asset.yml", type: null]
45    description: indicates that the physical-location is what is enclosed by (and
46    ↪ immediately surrounding) the asset, such as a
47      - building,
48      - structural tank,
```

```
44     - equipment cabinet,
45     - vehicle
46
47   class:
48
49     $ref: "../B-entity_class_object_schema/03_space_class.yml"
50
51   specification_data:
52
53     type: object
54     description: is a set of specification property data. (The applicable properties
55 ↪ are dependent on the definition made in the chosen class.)
56
57   inferred_classes:
58
59     oneOf:
60       - type: array
61         items:
62           $ref: "../B-entity_class_object_schema/03_space_class.yml"
63       - type: null
64     read-only: TRUE
65
66   service_address_or_coordinate:
67
68     oneOf:
69       - $ref: MaximoServiceAddressObject
70       - type: null
71     $comment: this is referencing Maximo's native service address object
72
73   status:
74
75     type: string
76     enum:
77       - specified
78       - realized
79       - eliminated
80     $comment: |
81 ↪ This field allows the user to specify whether the physical-location is merely
82 ↪ specified, or whether the boundary enclosing the physical-location have been
83 ↪ constructed (or alternatively, the fiat property / or area boundary around is
84 ↪ formally established and approved.) - i.e. "exists".
85
86     Note that an "eliminated" physical-location should be removed from the
87 ↪ hierarchy, and should not be visible for users conducting maintenance,
88 ↪ reliability, planning and scheduling functions.
89
90     The term "eliminated" is chosen to indicate that physical-location disappear
91 ↪ by the fact that object bound or defined the physical-location, such as wall,
92 ↪ ceilings, or property lines are removed.
93
94   confined_space:
95
96     type: boolean
97
98   inherit_hazardous_property_values:
```

```
89
90     type: boolean
91     default_value: true
92
93 hazardous location class:
94
95     type: string
96     enum:
97         - I
98         - II
99         - III
100
101 hazardous location division:
102
103     type: string
104     enum:
105         - 1
106         - 2
107
108 hazardous location group:
109
110     type: string
111     enum:
112         - A
113         - B
114         - C
115         - D
116         - E
117         - F
118         - G
119
120 # COMMON RECORD PROPERTIES
121
122
123 duplicate_record_of:
124
125     $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
126
127 record_retirement_information:
128
129     $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
130
```

04_org-chart_group.yml

```
1 ---
2 $schema: "http://json-schema.org/draft-07/schema#"
3 title: org-chart group
4 $id:
5   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/04_or
6 type: object
```

```
6 properties:
7
8   ID:
9
10     type: string
11     description: is the unique ID
12
13   org-chart_group_name:
14
15     type: string
16
17   parent_org-chart_group:
18
19     oneOf:
20       - $ref: "../04_org-chart_group.yml"
21       - type: null
22
23   class:
24
25     $ref: B-entity_class_object_schema/04_org_class.yml
26
27   specification_data:
28
29     type: object
30     description: is a set of specification property data. (The applicable properties
31 ↪ are dependent on the definition made in the chosen class.)
32
33   leader:
34
35     oneOf:
36       - $ref: "../07_person.yml"
37       - type: null
38     description:
39
40   equivalent_to_Maximo_site:
41
42     description: indicates that this org-chart group maps to a particular site (a
43 ↪ native Maximo object)
44     $ref: MaximoSiteObject
45
46   equivalent_to_Maximo_org:
47
48     description: indicates that this org-chart group maps to a particular org-chart
49 ↪ group (a native Maximo object)
50     $ref: MaximoOrgObject
51
52   equivalent_to_Maximo_crew:
53
54     description: indicates that this org-chart group maps to a particular org-chart
55 ↪ group (a native Maximo object)
56     $ref: MaximoCrewObject
57
58   # COMMON RECORD PROPERTIES
```

```
55
56
57 duplicate_record_of:
58
59   $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
60
61 record_retirement_information:
62
63   $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
64
65
66
```

05_item_master.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: item master
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/05_item_master.yml
6  type: object
7  properties:
8
9    ID:
10
11     description: A read-only UUID, generated by the system, to uniquely identify the
12     ↪ item.
13     type: string
14     implementer_note: |
15       Use UUID instead of a serial - this allow us to incorporate future items
16     ↪ defined outside of TW.
17
18     description:
19
20     description: A human readable short description of the item.
21     type: string
22     rule_spec:
23       - name: item master record naming
24         form: long
25         spec_ID: VkYgCtRPlx
26         type: assertion
27         specification: |
28           if (item_x.properties.commodity_or_commercial_product) = "commodity", then
29           ↪
30             the value of item_x.properties.name would be the semi-colon delimited
31           ↪ concatenation of the following property values:
32             - properties.class.properties.name
33             - every non-empty class dependent specification values
34             - properties.supplementary_commodity_description
35           elif: (item_x.properties.commodity_or_commercial_product = "commercial
36           ↪ product"), then:
```

```
32         the value of item_x.properties.name would be the semi-colon delimited
↪ concatenation of the following property values:
33         - properties.class.properties.name
34         - properties.product manufacturer company.properties."company name"
35         - properties.model_and_sub-model
36         - properties.version_or_model_year
37         - properties.product configuration code
38         checked on: 2024-08-15
39
40     class:
41
42     description: indicates the class to which this asset is an instance.
43     $ref: "../B-entity_class_object_schema/01_asset_item_tool_class.yml"
44     $comment: A value from the item classification (a superset of the asset class).
45     rule_spec:
46     - name: item classification list includes all classes
47       spec_ID: EynXVZ-dxg
48       specification: |
49           Maximo item classification list would include all class instances of
↪ B-entity_class_object_schema/01_asset_item_tool_class_object_schema.yml,
↪ regardless of the values
50       - properties.tool
51       - properties.only_used_as_a_part asset
52     status: specified
53     checked_on: 2024-08-15
54
55     rotating:
56
57     description: Signifies that instances of this item are tracked as serialized
↪ individuals.
58     $ref: "../00_common_definitions.yml#/definitions/rotating_property_def"
59
60     material_item:
61
62     description: indicates that the item is a material item, which are handled in
↪ continuous quantities.
63     type: boolean
64     $comment: |
65         Concept of material: materials are often handled in continuous quantities. For
↪ example, you might purchase a certain length of piping or a volume of concrete,
↪ and you have to measure and cut or shape these materials to fit your project's
↪ requirements.
66
67     # GENERIC COMMODITY AND COMMERCIAL PRODUCT DEFINITION
68
69
70     commodity_or_commercial_product:
71
72     description: Indicate whether the item master defines an unspecialized commodity
↪ or a specific commercial product of a certain manufacturer.
73     type: string
74     enum:
75     - commodity
```

```
76     - commercial product
77
78 # GENERIC COMMODITY DESCRIPTION
79
80
81 supplementary_commodity_description:
82
83     description: Supplementary description, in addition to the class value and
↪     class-dependent specification values, necessary to differentiate a commodity.
84     type: string
85     $comment: Toronto Water is not using the UNSPSC or any other commodity taxonomy
↪     code for item identification, as Toronto Water's classification system plays the
↪     same functional-location and can be mapped to other taxonomies.
86
87
88 # COMMERCIAL PRODUCT DESCRIPTION
89
90
91 commercial_product_definition:
92
93     description: Points to the commercial product definition.
94     oneOf:
95         - type: null
96         - $ref: "../00_common_definitions.yml#/definitions/manufacturer_and_model_def"
97
98 commercial_product_description:
99
100     description: A description concatenated from the text components of the
↪     commercial product.
101     oneOf:
102         - type: null
103         - type: string
104     rule_spec:
105         - name: Concatenation of commercial product description
106           status: TBD
107
108 instant_of_generic_commodity:
109
110     description: Indicates the commercial product is also a type of generic
↪     commodities.
111     oneOf:
112         - type: null
113         - type: array
114           items:
115             $ref: "../05_item_master.yml"
116
117 # OTHER ITEMS DESCRIPTIONS
118
119
120 item_format:
121
122     description: Describes the format of the individual units (forming the inventory
↪     count)- for example, "can" "functional-location", "sheet", "object", "box",
↪     "bag",....
```



```
123     type: string
124
125   specification_data:
126
127     type: object
128     description: is a set of specification property data. (The applicable properties
↪ are dependent on the definition made in the chosen class.)
129
130   cost_in_CAD:
131
132     description: The expected cost of an each unit of the item.
133     type: number
134
135   unit_of_issue:
136
137     description: Describes how the quantity of the item is measured, when it is
↪ issued out, such as "feet", "kg", "sheet".
138     type: string
139     rule_spec:
140       - name: Unit of issue for assets and parts (non-material items)
141         spec_ID: 01JK1VER5T6HK314XPB4W5T27V
142         type: validation
143         form: short
144         description: if the item is not an material item, its unit of issue would be
↪ "individual item"
145       - name: Default value of unit_of_issue
146         spec_ID: 01JK7AM2RA8S8EPAH57W33SJ79
147         type: assertion
148         form: short
149         description: When an item record is created, the default unit of issue is
↪ the same as the item format.
150
151   quantity_in_units_of_issue:
152
153     description: indicates how many units of issue is in the item.
154     type: number
155     rule_spec:
156       - name: Default quantity in units
157         spec_ID: 01JK7AFSHFQ2W8G5JD3B9XSWBV
158         type: assertion
159         form: short
160         description: When an item record is created, the default value of
↪ quantity_in_units_of_issue is 1.
161
162   alternate_format_of_same_item:
163
164     description: |
165       Usually identifies the same commercial product item made by the same
↪ manufacturer, but differing only in the format. For example, the 208-litre drum
↪ item and the 5-litre bottle item of Penzoil 5W30 Synthetic Lubricant.
166     oneOf:
167       - type: null
168       - type: array
```

```
169     items:
170       $ref: "./04_item_master.yml"
171   rule_spec:
172     - name: Range must be a commercial product as well
173       spec_ID: 01JF81079K178X9B4NSG23AA0Z
174       type: validation
175       specification:
176         status: TBD
177
178   ordering_options:
179
180   description: presents a list of vendors and available order formats
181   oneOf:
182     - type: null
183     - type: array
184       items:
185         type: object
186         properties:
187
188           vendor:
189
190             $ref: MaximoCompanyObject
191
192           unit_of_order:
193
194             type: object
195             properties:
196
197               description:
198
199                 description: a description the packages of individual units such
200 ↪ as "box of 24" or "individual unit"
201                 type: string
202
203               number_in_package:
204
205                 description: indicate the number of items that are in the package
206                 type: number
207
208               rule_spec:
209                 - name: Default order packaging format
210                   description: Default order packaging format is "individual unit"
211                   form: short
212                   spec_ID: 01JK1KEZSR9ASBXMWE406TCBPN
213
214           vendor_item_number:
215
216             oneOf:
217               - type: string
218               - type: null
219
220           cost_in_CAD:
```

```
221         description: The expected of the item in the order format
222         type: number
223         $comment: To TW, in the future, this field should contain a running
↪ average of the recent purchase costs, possibly also adjusted for recent
↪ inflation.
224
225     contracts:
226
227         description: the existing contracts that can be used for order the
↪ product
228         $ref: MaximoContractObject
229
230     # COMMON RECORD PROPERTIES
231
232
233     duplicate_record_of:
234
235         $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
236
237     record_retirement_information:
238
239         $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
```

06_tool_master.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: tool item master
4  $id:
↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/06_to
5  type: object
6
7  properties:
8
9      ID:
10     description: A read-only UUID, generated by the system, to uniquely identify the
↪ tool.
11     type: string
12     implementer_note: |
13         Use UUID instead of a serial - this allow us to incorporate future items
↪ defined outside of TW.
14
15     name:
16     type: string
17     description: The human readable short description of the tool.
18     rule_spec:
19     - name: tool master name
20       spec_ID: VJpSzGxdxg
21       type: implication
22       specification: |
23         if: toolX.properties."tool master type" = "generic tool"
```

```
24     then:
25         toolX.properties.name value is the semi-colon ("; ") delimited
↪ concatenation of the following property values:
26         - properties.class.properties."class name"
27         - properties."generic tool application definition"
28     elif: toolX.properties."tool type" = "specific commercial product"
29     then:
30         toolX.properties."tool name" value is the semi-colon ("; ") delimited
↪ concatenation of the following property values:
31         - properties."tool master class".properties."class name"
32         - properties.product manufacturer company.properties."company name"
33         - properties.model_and_sub-model
34         - properties.version_or_model_year
35         - properties.product configuration code
36     status: to be updated
37
38 class:
39     $ref: "../B-entity_class_object_schema/01_asset_item_tool_class.yml"
40     description: This is a value from the classification, which is a superset of the
↪ asset class.
41     rule_spec:
42         - name: Tool classification list does not include parts non tools
43           spec_ID: V1u1HHW0gx
44           specification: |
45             Tool classification list include all class instances of
↪ B-entity_class_object_schema/01_asset_item_tool_class_object_schema.yml, except
↪ ones whose .properties.tool value is FALSE
46           status: specified
47
48     specification_data:
49         type: object
50         description: is a set of specification property data. (The applicable properties
↪ are dependent on the definition made in the chosen class.)
51
52
53 # INVENTORY MANAGEMENT FLAGS
54
55 rotating:
56     $ref: "../00_common_definitions.yml#/definitions/rotating_property_def"
57
58 mobile:
59     type: boolean
60     description: An tool that is used beyond a permanent installation; instead, it
↪ is taken from place to place.
61     $comment: |
62         #PROCESS: SET default_value:
63         At record creation, set value to false.
64         #PROCESS: EVENT-DRIVEN VALUE CHANGE:
65         Upon the event of a properties.class value change;
66         if properties.class.properties."mobile" = true;
67         then set the value to true;
68         else set the value to false.
69
```

```
70 # COMMON RECORD PROPERTIES
71 #=====
72
73 duplicate_record_of:
74     $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
75
76 record_retirement_information:
77     $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
78
79 $comment: ASMP does not expect the tool item master to be widely used during the
    ↪ initial adoption of Maximo - we expect that most tools would initially be
    ↪ represented as un-stocked.
```

07_service_item_master.yml

```
1 $schema: "http://json-schema.org/draft-07/schema#"
2 title: tool item master
3 $id:
4   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/07_service_item_master.yml
5 type: object
6
7 properties:
8     # COMMON RECORD PROPERTIES
9
10    duplicate_record_of:
11
12        $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
13
14    record_retirement_information:
15
16        $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
17
18    # Rest to be specified in the future
```

08_person.yml

```
1 $schema: http://json-schema.org/draft-07/schema#
2 title: Person
3 $id:
4   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/08_person.yml
5 type: object
6
7 properties:
8     employee_number:
9         type: string
10
```

```
11 first_name:
12     type: string
13
14 last_name:
15     type: string
16
17 display_name:
18     description: Full name displayed (usually a combination of first and last names)
19     oneOf:
20         - type: null
21         - type: string
22
23 section:
24     description: Indicates which section within a division that the person works for
25     type: string
26     oneOf:
27         - type: null
28         - type: string
29
30 unit:
31     description: Indicates which business-unit within a section that the person
32 ↪ works for
33     oneOf:
34         - type: null
35         - type: string
36
37 status:
38     type: string
39     enum:
40         - active
41         - inactive
42         - departed
43
44 person_roles:
45     description: The role, associated with a certain type of system access, that the
46 ↪ person plays in the system.
47     oneOf:
48         - type: null
49         - type: array
50             items:
51                 $ref: MaximoSecurityRole
52
53 # NOTE: persons' association to trade in documented in the trade object
54
55 qualifications:
56     description: The qualifications that the person gained through training or
57 ↪ certification.
58     oneOf:
59         - type: null
60         - type: array
61             items:
```

```
61     type: object
62     properties:
63
64         qualification:
65             $ref: "../09_qualification.yml"
66
67         start_date:
68             description: the first effective day
69             oneOf:
70                 - type: null
71                 - type: string
72                 format: date
73
74         expiration_date:
75             description: the last effective date
76             oneOf:
77                 - type: null
78                 - type: string
79                 format: date
80
81     external_contractor:
82         description: Indicates whether the person is not an employee of the City
83         type: boolean
84
85     contract:
86         description: identifies the contracting company that the person works for
87         oneOf:
88             - type: null
89             - type: object
90             $ref: "../13_service_contract.yml"
91
92     contact:
93
94         properties:
95
96             email_address:
97                 description: Email address of the person
98                 type: string
99                 format: email
100
101             phone:
102                 description: Contact phone number of the person
103                 type: string
104
105     # COMMON RECORD PROPERTIES
106
107     duplicate_record_of:
108         $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
109
110     record_retirement_information:
111         $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
```

09_qualification.yml

```
1 $schema: http://json-schema.org/draft-07/schema#
2 title: Qualification
3 $id:
4   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/09_qualification.yml
5 type: object
6 properties:
7
8   ID:
9     description: is a read-only, unique, and permanent ID, generated by the system.
10    type: string
11    read-only: TRUE
12
13   name:
14     description: a short name given by the creator of the record.
15     type: string
16
17   status:
18     type: string
19     enum:
20       - active
21       - inactive
22
23   type:
24     type: string
25     enum:
26       - professional license
27       - skill certification
28       - record of training
29
30   issuing_organization:
31     oneOf:
32       - type: null
33       - $ref: "../04_org-chart_group.yml"
34
35 # COMMON RECORD PROPERTIES
36
37 duplicate_record_of:
38   $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
39
40 record_retirement_information:
41   $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
42
```

10_trade.yml

```
1 $schema: http://json-schema.org/draft-07/schema#
2 title: Trade
```



```
3 $id:
4   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/10_tr
5
6 type: object
7
8 properties:
9
10   code:
11     description: Unique identifier for the craft
12     type: string
13
14   name:
15     description: a short name given by the creator of the record.
16     type: string
17
18   organization:
19     description: a short name given by the creator of the record.
20
21   skill_levels:
22     type: array
23     description: List of skill levels within this craft
24     items:
25       type: object
26       properties:
27
28         skill_level:
29           type: string
30           description: Skill level associated with the craft (e.g., APPRENTICE,
31           ↪ JOURNEYMAN, MASTER)
32
33         hourly_rate:
34           description: Standard hourly rate for this skill level
35           type: number
36
37         qualifications:
38           description: qualifications required to hold
39           oneOf:
40             - type: null
41             - type: array
42               items:
43                 type: object
44
45   association_to_persons:
46     description: The primary craft or skill associated with the person, if the
47     ↪ person plays the role of a trade person.
48     oneOf:
49       - type: null
50       - type: array
51         items:
52           type: object
53           properties:
54
55           trade:
56             $ref: "../10_trade.yml"
```

```
53         skill_level:
54             type: string
55             $comment: load the list of skilled trades
56
57
58     status:
59         type: string
60         enum:
61             - active
62             - inactive
63
64 # COMMON RECORD PROPERTIES
65
66 duplicate_record_of:
67     $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
```

11_warranty.yml

```
1 $schema: http://json-schema.org/draft-07/schema#
2 title: Warranty
3 $id:
4   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/12_warranty.yml
5 type: object
6 properties:
7
8     ID:
9         description: the unique identifier for the warranty contract
10        type: string
11
12    description:
13        description: a brief summary of the warranty contract
14        type: string
15
16    vendor:
17        description: identifies the vendor or provider of the warranty
18        $ref: MaximoCompanyObject
19
20    warranty_start_date:
21        description: the first effective day of the warranty
22        oneOf:
23            - type: null
24            - type: string
25              format: date
26
27    warranty_expiration_date:
28        description: the last effective date of the warranty
29        oneOf:
30            - type: null
31            - type: string
32              format: date
```

```
33
34   covers_labour:
35     description: indicates that the vendor is responsible for providing and/or
↪    covering the cost of labour
36     type: boolean
37
38   covers_parts:
39     description: indicates that the vendor is responsible for providing and/or
↪    covering the cost of parts
40     type: boolean
41
42   covers_asset_functional-locations:
43     description: the list of functional-locations or more specifically the assets
↪    installed in the functional-locations covered by the warranty
44     oneOf:
45       - type: null
46       - type: array
47         items:
48           $ref: "../02_functional-location.yml"
49
50   covers_assets:
51     description: the list of assets covered by the warranty (any parts in the
↪    inventory covered by the warranty must be associated with an asset record)
52     oneOf:
53       - type: null
54       - type: array
55         items:
56           $ref: "../01_asset.yml"
57
58   contract_document:
59     description: the PDF version of the contract document.
60     oneOf:
61       - type: null
62       - type: array
63         items:
64           $ref: "../00_common_definitions.yml/attachment_def"
65
66 # COMMON RECORD PROPERTIES
67
68   duplicate_record_of:
69     $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
70
71   record_retirement_information:
72     $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
```

12_service_contract.yml

```
1 $schema: http://json-schema.org/draft-07/schema#
2 title: Service Contract
3 $id:
↪  https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/13_se
```

```
4  type: object
5
6  properties:
7
8    ID:
9      description: the unique identifier for the contract
10     type: string
11
12    description:
13      description: a brief summary of the contract
14      type: string
15
16    vendor:
17      description: identifies the vendor or provider
18      $ref: MaximoCompanyObject
19
20    start_date:
21      description: the first effective day
22      oneOf:
23        - type: null
24        - type: string
25          format: date
26
27    expiration_date:
28      description: the last effective date
29      oneOf:
30        - type: null
31        - type: string
32          format: date
33
34    covers_asset_functional-locations:
35      description: the list of functional-locations or more specifically the assets
36      ↪ installed in the functional-locations covered by the contract
37      oneOf:
38        - type: null
39        - type: array
40          items:
41            $ref: "./02_functional-location.yml"
42
43    covers_assets:
44      description: the list of assets covered by the contract (any parts in the
45      ↪ inventory covered by the contract must be associated with an asset record)
46      oneOf:
47        - type: null
48        - type: array
49          items:
50            $ref: "./01_asset.yml"
51
52    covers_work:
53      description: identifies work (job plans) the contractor would perform.
54      oneOf:
55        - type: null
56        - type: array
```

```
55     items:
56       $ref: "./32_job_plan.yml"
57
58   contract_document:
59     $ref: "./00_common_definitions.yml#/definitions/attachment_def"
60
61   # COMMON RECORD PROPERTIES
62
63   duplicate_record_of:
64     $ref: "./00_common_definitions.yml#/definitions/duplicate_record_def"
65
66   record_retirement_information:
67     $ref: "./00_common_definitions.yml#/definitions/record_retirement_def"
```

32_job_plan.yml

```
1  ---
2  $schema: http://json-schema.org/draft-07/schema#
3  title: Job Plan
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/32_job_plan.yml
6  description: |
7    A job plan is the the lowest level of work description to contain the full
8    ↪ planning specifications.
9  type: object
10
11  properties:
12    ID:
13      description: a read-only, unique, and permanent ID, generated by the system, to
14      ↪ identify the job plan record.
15      type: string
16
17    name:
18      description: a description of the activity specified in the job plan.
19      type: string
20
21    discrete_activity_classification:
22      description: indicates the type of activity that specified in the job plan
23      $ref: "../B-entity_class_object_schema/32_discrete_activity_class.yml"
24      $comment: this could also be called the activity classification
25
26    # JOB PLAN APPLICABILITY NOTES
27
28    specific_to_asset_classes:
29      description: identifies the asset classes on which the activity specified in
30      ↪ this job plan can be done.
31      type:
32        oneOf:
33          - type: null
```

```
32     - type: array
33       items:
34         $ref: "../B-entity_class_object_schema/01_asset_item_tool_class.yml"
35
36   specific_to_asset_functional-location_classes:
37     description: identifies the functional-location classes - more specifically, the
↪ assets installed in these functional-locations - for which this job plan is
↪ customized.
38     type:
39       oneOf:
40         - type: null
41         - type: array
42           items:
43             $ref:
↪ "../B-entity_class_object_schema/02_functional-location_classification.md"
44       $comment: >
45         Examples of a functional-location class include the tie-breaker and effluent
↪ turbidity meter functional-location class.
46
47   specific_to_operational_units:
48     description: identifies the Toronto Water site(s), defined as a org-chart group
↪ in this schema, for which the job plan is specifically customized.
49     type:
50       oneOf:
51         - type: null
52         - type: array
53           items:
54             $ref: "../A-entity_record_schema/04_org_class.yml"
55     $comment: >
56       Examples of operational group include
57       TAB - Ashbridges Bay Wastewater Treatment Plants
58       COL - Waste and Storm Water Collection
59     rule_spec: >
60       - name: job plan's specific_to_operational_units property must be an unit
↪ level org-chart group
61       spec_ID: 01JD2V5X97J1Y45JWDW4SV1FJ4
62       type: [validation]
63       specification: |
64         Given an job plan, JP_x, all values of JP_x.specific_to_operational_units
↪ must be
65         - an org-chart group, and
66         - whose .class = unit (a subclass of Group in the City)
67
68   specific_to_asset_functional-locations:
69     description: identifies the functional-locations for which the job plan is
↪ specifically customized.
70     type:
71       oneOf:
72         - type: null
73         - type: array
74           items:
75             $ref: "../A-entity_record_schema/02_functional-location.yml"
76
```

```
77 specific_to_commercial_products:
78   description: identifies the the commercial products for which the job plan is
↪ specifically written/customized.
79   type:
80     oneOf:
81     - type: null
82     - type: array
83       items:
84         $ref: "../A-entity_record_schema/05_item_master.yml"
85   rule_spec: >
86     - name: Valid Commercial Product Item reference in a Job Plan
87       spec_ID: 01JD2V5X97J1Y45JWDW4SV1FJ4
88       type: [validation]
89       specification: |
90         Given an job plan, JP_x, all values of
↪ JP_x.specific_to_commercial_products must be
91         - a item master record, and
92         - whose .generic_or_specific_product = "specific commercial product"
93
94 # RECORD PROVENANCE
95
96 is_derived_from:
97   description: identifies the job plan from which the present job plan
↪ specification was based on.
98   oneOf:
99   - type: null
100   - $ref: "../32_job_plan.yml"
101   integration: PM library
102
103 failure_codes:
104   description: denotes a physical-based failure condition (e.g., shaft
↪ misalignment).
105   oneOf:
106   - type: null
107   - type: array
108     items:
109       $ref: "../00_common_definitions.yml#/definitions/failure_code"
110
111 RCM_failure_modes:
112   description: identifies the functional failure mode ID, with respect to a
↪ specific functional-location, mitigated by the work specified in this job plan.
113   oneOf:
114   - type: null
115   - type: array
116     items:
117       type: string
118   integration: PM Library
119   $comment: >
120     - For the 2026-27 implementation, this data field will start-out as a
↪ free-text. In the future, the value will come from a solution such as OnePM.
121
122 # PROCEDURE
123
```

```
124   work_description:
125     description: is a single body of text outlining the sequential steps to complete
↪   the activity
126     type: string
127     $comment: >
128       Example:
129         1) Ensure you have operational approval before performing this task.
130         2) Follow Lock-out and Tag-out process before starting this task.
131         3) Drain the oil from the gearbox.
132         4) Install 25 Litres of UCON 220 ( food grade) oil.
133         5) Remove Lock-out and Tag-out and check operation.
134         6) Inform operations that the task as assigned is completed.
135
136   requires_shut_down:
137     description:
138     type: boolean
139
140 # RESOURCE REQUIREMENTS
141
142   estimated_duration:
143     description: is the estimated time to complete the activity in the job plan
144     $ref: "../00_common_definitions.yml#/definitions/frequency_interval_definition"
145
146   parts_or_material_requirements:
147     description: identifies the parts and material required to complete the work.
148     oneOf:
149       - type: null
150       - type: array
151         items:
152           $ref:
↪   "../00_common_definitions.yml#/definitions/item_requirement_definition"
153
154   tool_requirements:
155     description: identifies the tools required to complete the work.
156     oneOf:
157       - type: null
158       - type: array
159         items:
160           $ref:
↪   "../00_common_definitions.yml#/definitions/tool_requirements_definition"
161
162   skill_and_trade_requirements:
163     description: identifies the trades and qualifications of each trade needed to
↪   complete the work.
164     oneOf:
165       - type: null
166       - type: array
167         items:
168           $ref: "../00_common_definitions.yml#/definitions
↪   trade_requirement_definition"
169
170   service_requirements:
171     description: identifies (contracted) service needed to complete the work.
```



```
172     oneOf:
173       - type: null
174       - type: array
175         items:
176           $ref:
↪     "./00_common_definitions.yml#/definitions/service_requirement_definition"
177
178 # RELATED ACTIVITIES
179
180 must_be_preceded_by:
181   description: identifies activities (specified in other job plans) that must be
↪   performed in the same work order before the activity specified in this PM can be
↪   performed.
182   oneOf:
183     - type: null
184     - type: array
185       items:
186         $ref: "./32_job_plan.yml"
187
188 must_be_followed_by:
189   description: identifies activities (specified in other job plans) that must be
↪   performed in the same work order after the activity specified in this PM can be
↪   performed.
190   oneOf:
191     - type: null
192     - type: array
193       items:
194         $ref: "./32_job_plan.yml"
195
196 # WORK TRIGGER CONDITION NOTES
197
198 time-based_frequency:
199   oneOf:
200     - type: null
201     - $ref:
↪     "./00_common_definitions.yml#/definitions/frequency_interval_definition"
202
203 meter-based_frequency:
204   oneOf:
205     - type: null
206     - $ref: "./00_common_definitions.yml#/definitions/meter_condition_definition"
207
208 description_of_event-based_trigger:
209   oneOf:
210     - type: null
211     - type: object
212       properties:
213
214         relation_to_event:
215
216           type: string
217           enum:
218             - before
```

```
219         - during
220         - after
221         - at the start of
222         - at the end of
223
224     description_of_event:
225
226         description: a free-text description of a event or process, such as "an
↪ elevator failure".
227         type: string
228
229     notes_on_trigger_condition:
230         description: free-text description on the additional trigger conditions
231     oneOf:
232         - type: null
233         - type: string
234
235 # COMPLIANCE INFORMATION
236
237     compliance_requirement:
238         description: identifies the compliance requirement object
239         integration: data-hub
240         oneOf:
241             - type: null
242             - type: array
243             items:
244                 $ref: "./00_common_definitions.yml#/definitions/compliance_requirement"
245
246     compliance_class:
247         description: indicates the level of compliance, with legislative being the top
↪ class
248         $ref: "./00_common_definitions.yml#/definitions/compliance_class"
249
250     mitigates_safety_risk_to_staff:
251         oneOf:
252             - type: null
253             - type: string
254         enum:
255             - yes
256             - no
257             - unspecified
258
259     mitigates_safety_or_health_risk_to_public:
260         oneOf:
261             - type: null
262             - type: string
263         enum:
264             - yes
265             - no
266             - unspecified
267
268     mitigates_environmental_risk:
269         oneOf:
```

```
270   - type: null
271   - type: string
272   enum:
273     - yes
274     - no
275     - unspecified
276
277   safe_work_plan_link:
278     description: identifies a safety work plan by a permanent URL to the document
↪    (e.g., corporate Safe Procedure or Toronto Water Safe Operating Procedures)
279     oneOf:
280       - type: null
281       - type: array
282         items: #URL strings
283           type: string
284
285 # COMMON RECORD PROPERTIES
286
287 duplicate_record_of:
288   $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
289
290 record_retirement_information:
291   $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
```

33_PM.yml

```
1  ---
2  $schema: http://json-schema.org/draft-07/schema#
3  title: PM
4  $id:
↪   https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/33_PM
5  type: object
6
7  properties:
8
9    ID:
10     description: is a read-only, unique, and permanent ID, generated by the system,
↪    to identify PM.
11     type: string
12     $comment: This ID is useful for referencing, even when its name changes.
13
14    name:
15     description: a short name for the PM, given by the creator of the PM.
16     type: string
17
18    operational_unit:
19     description: indicates Toronto Water's operational unit, on the org-chart group
↪    hierarchy, such as Waste and Storm Water Pumping (symbol - WASP) or Humber
↪    Wastewater Treatment Plant (symbol - THR).
20     $ref: "../04_org-chart_group.yml"
21
```

```
22 supervisor_group:
23   description: indicates a sub-group of the operational unit, that is led by a
↪ supervisor who is accountable for the performance of the specified work.
24   $ref: "./04_org-chart_group.yml"
25
26 crew_assignment:
27   description: identifies a crew, under the supervisor group, that is always
↪ assigned to perform the specified work
28
29   oneOf:
30     - type: null
31     - $ref: "./04_org-chart_group.yml"
32
33   comment: This value will be determined by (and must be consistent with) the
↪ maintainer_org-chart_group value - situated at a lower level of the org-chart
↪ group hierarchy
34
35 member_of_PM_set:
36   description: indicates that this PM is a member of a set of related PMs. For
↪ example, the PMs for raw water pump 1, 2, 3 are all members of a PM set named
↪ Raw Water 5-year Disassembly Maintenance.
37   oneOf:
38     - type: null
39     - $ref: "#/definitions/PM_set"
40   rule to add []: only applicable to higher-level PM
41
42 Avantis_PM:
43   description: indicate the Avantis PM (the legacy WMS) that this Maximo PM
↪ (equivalent to an Avantis PM task) was a part of.
44   oneOf:
45     - type: null
46     - type: string
47   read-only: TRUE
48   comment: This field can be eliminated in the future.
49   work_entity_harmonization: WR(x), WO(x), JP(x)
50
51 processes_covered_by_PM:
52   description: a list of all major process systems covered by the work specified
↪ in this PM.
53   read-only: TRUE
54   oneOf:
55     - type: null
56     - type: object
57       properties:
58         ranking:
59           type: number
60         system_naming:
61           type: string
62   rule to add []: only applicable to higher-level PM
63
64 # WORK SPECIFICATION AT A HIGH-LEVEL
65
66 functional_location_to_work_on:
```

```
67     description: indicates functional-location at which the specified must be
↪ performed.
68     oneOf:
69       - type: null
70       - $ref: "../A-entity_record_schema/02_functional-location.yml"
71
72     asset:
73       description: indicates the asset that is being maintained.
74       oneOf:
75         - type: null
76         - $ref: "../A-entity_record_schema/01_asset.yml"
77
78     job_plan:
79       description: specifies the job plan for the PM, if there is no further
↪ specification within the route.
80
81 # PM STRUCTURE SPECIFICATION
82
83     parent:
84       description: indicates the more comprehensive PM, usually a shut-down PM, that
↪ this PM is a part of.
85       oneOf:
86         - type: null
87         - $ref: "../A-entity_record_schema/33_PM.yml"
88       $comment: PMs should be organized into a PM-set when they are meant to be
↪ performed at different times. For example the PMs for substation line 1 and
↪ line 2 maintenance are performed on alternating years. They can be organize into
↪ a PM-Set named Main Substation Maintenance. PMs should be organized under a
↪ parent PM if they are parts of the same larger continuous process - represented
↪ by the parent - such as the winter shutdown maintenance of island treatment
↪ plant.
89
90     route:
91       description: a sequential list of work, composed of job plans paired with an
↪ asset/functional-location.
92       oneOf:
93         - type: null
94         - type: object
95           properties:
96             sequence:
97               type: number
98             asset:
99               oneOf:
100                 - type: null
101                 - $ref: "../01_asset.yml"
102             functional-location:
103               oneOf:
104                 - type: null
105                 - $ref: "../02_functional-location.yml"
106             job_plan:
107               oneOf:
108                 - type: null
109                 - $ref: "../32_job_plan.yml"
```

```
110     $comment: the implementation could be done with Maximo route object.
111
112 # RESOURCES
113
114 #Note:
115     # content: Travel time and preparation time are not being recorded explicitly on
↪ the PM. Instead they could be recorded as contributory work in the job plan
↪ route
116
117     estimated_duration:
118         description: is the estimated time to complete the activity in the job plan
119         $ref: "./00_common_definitions.yml#/definitions/frequency_interval_definition"
120
121     parts_or_material_requirements:
122         description: identifies the parts or material required to complete a work order
↪ generated from the PM.
123         oneOf:
124             - type: null
125             - type: array
126                 items:
127                     $ref:
↪ "./00_common_definitions.yml#/definitions/item_requirement_definition"
128
129     tool_requirements:
130         description: identifies the tools required to complete a work order generated
↪ from the PM.
131         oneOf:
132             - type: null
133             - type: array
134                 items:
135                     $ref:
↪ "./00_common_definitions.yml#/definitions/tool_requirements_definition"
136
137     skill_and_trade_requirements:
138         description: identifies the trades and qualifications of each trade needed to
↪ complete the work.
139         oneOf:
140             - type: null
141             - type: array
142                 items:
143                     $ref: "./00_common_definitions.yml#/definitions
↪ trade_requirement_definition"
144
145     service_requirements:
146         description: identifies (contracted) service needed to complete to complete a
↪ work order generated from the PM.
147         oneOf:
148             - type: null
149             - type: array
150                 items:
151                     $ref:
↪ "./00_common_definitions.yml#/definitions/service_requirement_definition"
152
```

```
153 # Work Triggering Specification
154
155 # Note: the specification is not complete for data mapping purposes, it is
↪ complete for requirement gather
156
157 next_due_date_based_on:
158   type: string
159   enum:
160     - work start date
161     - work completion date
162
163 trigger_condition: # aka work generation condition
164   oneOf:
165     - type: null
166     - $ref: "#/definitions/time-based_trigger_specification"
167     - $ref: "#/definitions/meter-based_trigger_specification"
168
169
170 # LEGISLATIVE DESIGNATION
171
172 compliance_level:
173   $ref: "./00_common_definitions.yml#/definitions/compliance_class"
174   work_entity_harmonization: WR(x), WO(_), JP(_)
175
176
177 # COMMON RECORD PROPERTIES
178
179 duplicate_record_of:
180   $ref: "./00_common_definitions.yml#/definitions/duplicate_record_def"
181
182 record_retirement_information:
183   $ref: "./00_common_definitions.yml#/definitions/record_retirement_def"
184
185 #####
186 # LOCAL OBJECT DEFINITIONS
187 #####
188
189 definitions:
190
191   time-based_trigger_specification:
192     next_due_date:
193       type: string
194       format: date
195     frequency_interval:
196       $ref: "./00_common_definitions.yml#/definitions/frequency_interval_definition"
197
198   meter-based_trigger_specification:
199     reading_at_last_service_completion:
200       type: number
201       description: is the last recorded meter reading when service was completed.
202     next_service_trigger_reading:
203       type: number
204
```

```
205     meter_condition:
206       oneOf:
207         - type: null
208         - $ref:
↪     " ../00_common_definitions.yml#/definitions/meter_condition_definition"
```

34_FR_WR_WO.yml

```
1  $schema: http://json-schema.org/draft-07/schema#
2  title: Failure Report, Work Request, Work Order
3  $id:
↪   https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/34_FR
4
5  $comment: In this design, the work request doubles as a failure report.
6
7  properties:
8
9     ID:
10
11     description: is a read-only unique ID, generated by the system, to uniquely
↪     identify the record.
12     type: string
13
14     implementation:
15       MX_mapping: WONUM
16
17     311_ticket_ID:
18
19     implementation:
20       MX_mapping: COTTICKETID
21       D&C_only: true
22
23     311_request_number:
24
25     implementation:
26       MX_mapping:
27       D&C_only: true
28
29     $comment: |
30       [] to resolve: there is some uncertainty of this should be either
↪     COTREQUESTNUMBER or EXTERNALREFID
31
32     record_type:
33
34     description: indicates whether this work request is being used to track an
↪     failure, without being a request for work to address the failure.
35     type: string
36     enum:
37       - failure report
38       - work request
39       - work order
```



```
40
41   rule_spec:
42     - name: inference of current_type from status
43       status: []
44
45   status:
46
47     description: indicate the status of failure report, work request, and work order
48     type: string
49     enum:
50       - failure reported
51       - request made
52       - request approved
53       - request cancelled
54       - waiting on resource
55       - ready to schedule
56       - scheduled
57       - in progress
58       - completed
59       - closed
60       - WO cancelled
61
62     implementation:
63       MX_mapping: WO.status
64
65   following-up_on:
66
67     description: identifies the work order which this record is following up on.
68     read-only: true
69     oneOf:
70       - type: null
71       - $ref: "../A-entity_record_schema/35_work_order.yml"
72
73   # THE OBJECT
74
75   functional-location_to_work_on:
76
77     description: indicates functional-location at which the specified work is to be
78     ↪ performed.
79     oneOf:
80       - type: null
81       - $ref: "../A-entity_record_schema/02_functional-location.yml"
82       #[] should have name and description, like asset_to_work_on
83
84   rule_spec:
85     - name: mutual exclusion of asset_to_work_on and
86     ↪ functional-location_to_work_on values
87       spec_ID: 01JFK49T43T1TF3HBTBTDPMN5Y
88       form: short
89       specification: |
90         one, but only one, of the following properties can have a non-null value:
91         - functional-location_to_work_on
92         - asset_to_work_on
```

```
91
92 asset_to_work_on:
93
94   description: indicates asset on which the specified work is to be performed.
95   oneOf:
96     - type: null
97     - $ref: "../A-entity_record_schema/01_asset.yml"
98     properties:
99       ID:
100
101       implementation:
102         MX_mapping: WO.ASSETNUM
103       name:
104
105
106   implementation:
107     MX_mapping: WO.ASSETNUM
108     $comment: |
109       multiple fields may need to represented.
110
111 asset_or_asset_functional-location_criticality:
112
113   description: is the criticality values inherited from the asset or
114 ↪ functional-location being worked on.
115   read-only: true
116   oneOf:
117     - type: null
118     - type: object
119     properties:
120
121       rating:
122         type: number
123
124       description:
125         type: string
126
127 issue_found_at_address:
128
129   description: indicates the service address in or around which a failed asset
130 ↪ requiring maintenance is located.
131
132   oneOf:
133     - type: null
134     - $ref: MaximoServiceAddressObject
135
136 route:
137
138   description: a sequential list of work, composed of job plans paired with an
139 ↪ asset/functional-location.
140   oneOf:
141     - type: null
142     - type: object
143     properties:
```

```
141
142     sequence:
143         type: number
144
145     asset:
146         oneOf:
147             - type: null
148             - $ref: "./01_asset.yml"
149
150     functional-location:
151         oneOf:
152             - type: null
153             - $ref: "./02_functional-location.yml"
154
155     job_plan:
156         $ref: "./32_job_plan.yml"
157
158     $comment: the implementation could be done with Maximo route object.
159
160
161 # FAILURE REPORTING
162
163
164 description_of_issue:
165
166     description: is a factual description of the observable aspects of a single
167 ↪ issue..
168     type: string
169
170     implementation:
171         WR_only: true
172
173     observed_problems:
174
175     description: is what's also known as a common symptom code (e.g., making noise,
176 ↪ cannot start, not running).
177     oneOf:
178         - type: null
179         - type: array
180           items:
181             $ref: "./00_common_definitions.yml#/definitions/failure_code"
182
183     implementation:
184         WR_only: true
185
186     rule_spec:
187         - name: Which problem codes to show
188           spec_ID: 01JFH3ERR08WHJOE4WRK166WRT
189           form: short
190           specification: []
191
192     $comment: multiple values are allowed
```

```

192 physical_causes:
193
194   oneOf:
195     - type: null
196     - type: array
197       items:
198         properties:
199           cause_code:
200             description: denotes a physical-based failure condition (e.g., shaft
↪ misalignment).
201             $ref: "../00_common_definitions.yml#/definitions/failure_code"
202             $comment: one failure code per request
203             rule_spec:
204               - name: Which Cause Codes to Show
205                 spec_ID: 01JFH2F04P28B4EB2HNWA68KN9
206                 form: short
207                 specification: |
208                   - type must be "cause"
209                   - [] more tbd
210             action_spec:
211               - name: Creating a New Failure Code
212                 spec_ID: 01JFH2NE68WONCSBKSYZRA2Z
213                 form: short
214                 specification: User must be able to specify new failure codes,
215             basis_of_selection:
216               description: indicates how the failure code was derived
217               type: string
218               enum:
219                 - actual observation
220                 - educated guess from signs
221             implementation:
222               WR_only: true
223
224   bread_crumb:
225
226   found_asset_offline_due_to_this_failure:
227
228     description: indicates whether the asset was offline, because of the failure
229     type: boolean
230     implementation:
231       WR_only: true
232     $comment: if true, failure reporting is required []Rule
233
234   took_asset_offline_due_to_this_failure:
235
236     description: indicates whether the asset had to be taken offline, because of the
↪ failure
237     type: boolean
238     implementation:
239       WR_only: true
240
241
242   # WORK DETAIL

```

```
243
244
245   work_title:
246
247     description: a short text summarizing the work that is being requested or have
↪   been approved to be performed.
248     oneOf:
249       - type: null
250       - type: string
251
252     implementation:
253       MX_mapping: WO.description
254
255   work_specification:
256
257     description: a sufficiently detailed description of the work being requested for
↪   the approver of the work.
258
259     implementation:
260       MX_mapping: WO.DESCRPTION_LONGDESCRIPTION
261
262   work_priority:
263
264     description: a synthetic number derived from the condition of the asset function
↪   being maintained (i.e., how close is it to failure), and the importance (or
↪   criticality) of the asset function to the org-chart group's goals.
265
266     implementation:
267       MX_mapping: INTERNALPRIORITY
268
269   job_plan:
270
271     description: specifies the job plan for the PM, if there is no further
↪   specification within the route.
272
273     oneOf:
274       - type: null
275       - type: object
276         $ref: "../A-entity_record_schema/32_job_plan.yml"
277
278     action_spec:
279       name: Importing Specifications from a Job Plan
280       form: short
281       id: 01JFVCVT6Q5F62WAHEB001J7SX
282       specification: TBD []
283
284   work_type:
285
286     description: is the classification at the work order level
287     $ref: "../B-entity_class_object_schema/33_work_type.yml"
288
289     rule_spec:
290       name: Failure Reporting Leads to Investigation or Repair
```

```
291     form: short
292     id: 01JFVCZ9Y7G5MWP2G2DADB2G8Z
293     specification: if failure is reported, then work type must either be
↪    investigative or repair
294
295     discrete_activity_classification:
296
297     description: is a classification often inherited from the job plan specified on
↪    the work order
298     $ref: "../B-entity_class_object_schema/33_work_type.yml"
299     not_on_WR: true
300
301     site:
302
303     description: indicates Toronto Water's operational unit, on the org-chart group
↪    hierarchy, such as Waste and Storm Water Pumping (symbol - WASP) or Humber
↪    Wastewater Treatment Plant (symbol - THR).
304     $ref: "../04_org-chart_group.yml"
305
306     implementation:
307         $comment: WO.SITEID
308
309     maintenance_group:
310
311     description: indicates a sub-group of the operational unit, that is led by a
↪    supervisor who is accountable for performing the work.
312     $ref: "../04_org-chart_group.yml"
313
314     rule_spec:
315         name: Inherit the maintenance_group value from either the asset or the
↪    functional-location
316         spec_ID: 01JFK43CJBC495TB7Y3H3VP172
317         form: very short
318
319     requires_asset_offline:
320
321     description: indicates the work requires the asset to be offline
322     oneOf:
323         - type: null
324         - type: string
325     enum:
326         - yes
327         - no
328         - unknown
329
330     rule_spec:
331         - name: Default value of requires_asset_offline is null
332           req_spec_ID: 01JFK2JOHWVWKDK4WWK5RZCXWY
333           form: very short
334
335
336     # RESPONSIBILITIES AND ASSIGNMENTS
337
```

```
338 crew_assignment:
339
340   description: identifies a crew, under the supervisor group, that is always
↪ assigned to perform the specified work
342
343   oneOf:
344     - type: null
345     - $ref: "../04_org-chart_group.yml"
346
347   WO_only: true
348
349   comment: This value will be determined by (and must be consistent with) the
↪ maintainer_org-chart_group value - situated at a lower level of the org-chart
↪ group hierarchy
350
351 trades_assignment:
352
353   description: identifies the individual trades-persons who will be performing the
↪ work order.
354   oneOf:
355     - type: null
356     - type: array
357       items:
358         $ref: "../08_person.yml"
359
360 asset_covered_by_warranty_contract:
361
362   description: indicates that the asset (or the asset in the functional-location)
↪ is currently covered by a warranty contract.
363   type: boolean
364
365   implementation:
366     MX_mapping: WO.WARRANTYEXIST
367
368   todo []: rule - determine the value from the asset's warranty information.
369
370 warranty_expiration_date:
371
372   description: indicates the date the warranty expires, if the asset is covered by
↪ a warranty contract.
373   type: string
374   format: date
375
376   implementation:
377     MX_mapping: WO.WARRANTYEXPDATE
378
379 send_work_to_warranty_contractor:
380
381   description: a true (or yes) value indicates that the specified work should be
↪ performed by the warranty contractor.
382   type: boolean
383   todo []: rule - enable this field, only if asset_covered_by_warranty_contract is
↪ true
```

```
384
385   asset_covered_by_service_contract:
386
387     description: indicates that the asset (or the asset in the functional-location)
↪   is currently covered by a service contract.
388     type: boolean
389
390   send_work_to_service_contractor:
391
392     description: a true (or yes) value indicates that the specified work should be
↪   performed by a the selected service contractor.
393     oneOf:
394       - type: null
395       - $ref: "../A-entity_record_schema/07_service_item_master.yml"
396
397   supports_a_capital_project:
398
399     description: indicate that the specified work supports the work being done be a
↪   capital project consultant or contractor.
400     type: boolean
401
402   supports_the_capital_project:
403
404     description: indicates the specific capital project (represented as a work
↪   order)
405     oneOf:
406       - type: null
407       - $ref: "../A-entity_record_schema/35_work_order.yml"
408
409
410   # RESOURCES
411
412
413   estimated_duration:
414
415     description: is the estimated time required in hours to complete the activity in
↪   the job plan
416     oneOf:
417       - type: null
418       - type: number
419
420   part_or_material_requirements:
421
422     description: identifies the parts or material required to complete a work order
↪   generated from the PM.
423     oneOf:
424       - type: null
425       - type: array
426         items:
427           $ref:
↪   "../00_common_definitions.yml#/definitions/item_requirement_definition"
428
429   tool_requirements:
```



```
430
431   description: identifies the tools required to complete a work order generated
↪   from the PM.
432   oneOf:
433     - type: null
434     - type: array
435       items:
436         $ref:
↪   ".00_common_definitions.yml#/definitions/tool_requirements_definition"
437
438   service_requirements:
439
440   description: identifies (contracted) service needed to complete to complete a
↪   work order generated from the PM.
441   oneOf:
442     - type: null
443     - type: array
444       items:
445         $ref:
↪   ".00_common_definitions.yml#/definitions/service_requirement_definition"
446
447   skill_and_trade_requirements:
448
449   description: identifies the trades and qualifications of each trade needed to
↪   complete the work.
450   oneOf:
451     - type: null
452     - type: array
453       items:
454         $ref: ".00_common_definitions.yml#/definitions
↪   trade_requirement_definition"
455
456
457   # DATES
458
459
460   issue_reported_date:
461
462   description: is the date that the issue or failure was reported.
463   oneOf:
464     - type: null
465     - type: string
466       format: date
467
468   work_requested_date:
469
470   description: is the date when the work request was submitted.
471   oneOf:
472     - type: null
473     - type: string
474       format: date
475
476   request_approval_date:
```

```
477
478     description: is the date that the work request was approved (and when it became
↪ a work order).
479     oneOf:
480       - type: null
481       - type: string
482         format: date
483
484     target_start_date:
485
486     description: is the date when the work should begin (according to a certain
↪ service standard).
487     oneOf:
488       - type: null
489       - type: string
490         format: date
491
492     target_completion_date:
493
494     description: is the date when the work should be completed (according to a
↪ certain service standard).
495     oneOf:
496       - type: null
497       - type: string
498         format: date
499
500     scheduled_start_date:
501
502     description: is the date when the work is scheduled (by a scheduler) to begin.
503     oneOf:
504       - type: null
505       - type: string
506         format: date
507
508     scheduled_completion_date:
509
510     description: is the date when the work is scheduled (by a scheduler) to be
↪ completed.
511     oneOf:
512       - type: null
513       - type: string
514         format: date
515
516     actual_start_date:
517
518     description: is the date when the work actually began.
519     oneOf:
520       - type: null
521       - type: string
522         format: date
523
524     actual_completion_date:
525
```

```
526     description: is the date when the work was actually completed.
527     oneOf:
528       - type: null
529       - type: string
530         format: date
531
532     cancel_date:
533
534     description: is the date when the work was cancelled (and the record became
↪ either a cancelled work order or work request).
535     oneOf:
536       - type: null
537       - type: string
538         format: date
539
540     WO_closing_date:
541
542     description: is the date when the work order was closed.
543     oneOf:
544       - type: null
545       - type: string
546         format: date
547
548
549     # FAILURE INFORMATION INHERITED
550
551
552     member_of_PM_set:
553
554     description: indicates that this PM is a member of a set of related PMs. For
↪ example, the PMs for raw water pump 1, 2, 3 are all members of a PM set named
↪ Raw Water 5-year Disassembly Maintenance.
555
556     oneOf:
557       - type: null
558       - $ref: "#/definitions/PM_set"
559
560     parent_work_order:
561
562     description: indicates the more comprehensive PM, usually a shut-down PM, that
↪ this PM is a part of.
563     oneOf:
564       - type: null
565       - $ref: "../A-entity_record_schema/33_PM.yml"
566
567     WO_only: true
568
569     rule_spec:
570       name: Work Type of Descendant Work Orders
571       form: short
572       id: 01JFVDM89RVDCE7VBVM7FDQHRD
573       specification: In a work order hierarchy, the top-level work order determines
↪ the work type of all descendant work orders.
```

```
574
575   $comment: |
576     PMs should be organized into a PM-set when they are meant to be performed at
↪   different times. For example the PMs for substation line 1 and line 2
↪   maintenance are performed on alternating years. They can be organize into a
↪   PM-Set named Main Substation Maintenance. PMs should be organized under a parent
↪   PM if they are parts of the same larger continuous process - represented by the
↪   parent - such as the winter shutdown maintenance of island treatment plant.
577
578   summary_of_previous_issue_reports:
579
580     description: presents a summary of previously reported issues and failures
↪   related to this work.
581     oneOf:
582       - type: null
583       - type: object
584         name: compiled_issue_report
585         properties:
586           compiled_text_summary:
587             description: the compilation of all text information in a issue report,
↪   including problem code, failure code, and description.
588             oneOf:
589               - type: null
590               - type: array
591                 items:
592                   type: string
593
594           photographs:
595             description: photographs in the failure report.
596             oneOf:
597               - type: null
598               - type: array
599                 items:
600                   $ref: "./00_common_definitions.yml/attachment_def"
601
602   mitigates_safety_risk_to_staff:
603
604     description: indicates the work has impact on workers' safety
605     oneOf:
606       - type: null
607       - type: string
608     enum:
609       - yes
610       - no
611       - unspecified
612
613   mitigates_safety_or_health_risk_to_public:
614
615     description: indicates that the work has a direct impact on the well-being of
↪   the public
616     oneOf:
617       - type: null
618       - type: string
```

```
619   enum:
620     - yes
621     - no
622     - unspecified
623
624   mitigates_environmental_risk:
625
626     description: indicates that the work has impact on environmental protection
627     oneOf:
628     - type: null
629     - type: string
630     enum:
631       - yes
632       - no
633       - unspecified
634
635   # LEGISLATIVE DESIGNATION
636
637   compliance_class:
638
639     description: indicate that the completion of the specified work would satisfy
640 ↪ some compliance requirement of a certain Level.
641     $ref: "../00_common_definitions.yml#/definitions/compliance_class"
642
643   attachments:
644
645     description: documents or photographs that provide further supplementary
646 ↪ information.
647     oneOf:
648     - type: null
649     - type: array
650       items:
651         $ref: "../00_common_definitions.yml/attachment_def"
652
653     implementation:
654       MX_mapping:
655
656   # COMMON RECORD PROPERTIES
657
658   duplicate_record_of:
659
660     $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
661
662   record_retirement_information:
663
664     $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
```

36_work_order_documentation.yml

```
1  ---
2  $schema: http://json-schema.org/draft-07/schema#
3  title: Work Order Documentation
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/1-Schemas/A-entity_record_schema/36_work_order_documentation.yml
6  type: object
7  properties:
8    work_order_reference:
9      description: identifies the work order, whose actual that is being documented.
10     read-only: true
11     $ref: "../A-entity_record_schema/35_work_order.yml"
12
13  # OPERATIONAL STATUS INFORMATION
14
15  #Note:
16    # content: in a work order containing children work orders, these information
17    ↪ only have to be filled in at the parent level.  [] rule
18
19  asset_offline_at_start:
20    description: indicates that the asset was offline when the work began
21    type: boolean
22    $comment:
23
24  asset_brought_back_online:
25    description: is only applicable if the asset was offline when the work began;
26    ↪ this entry indicates that the work, within the scope of this work order, brought
27    ↪ the asset back online
28    type: boolean
29    $comment:
30
31  # ACTUAL WORK ORDER RESOURCE USAGE
32
33  revised_description_of_actual_work:
34    description: is a revised and more accurate description of the actual work
35    ↪ performed on the asset.
36    type: string
37
38  actual_start_time:
39    description: the date the work started
40    oneOf:
41      - type: null
42      - type: string
43        format: date
44
45  actual_completion_time:
46    description: the date the work was actually completed
47    oneOf:
48      - type: null
49      - type: string
```

```
46     format: date
47
48   actual_wrench_time:
49     description: is the actual time taken to complete the work order.
50     $ref: "./00_common_definitions.yml#/definitions/frequency_interval_definition"
51
52   actual_parts_and_material_usage:
53     description: identifies the parts and material used in completing the work.
54     oneOf:
55       - type: null
56       - type: array
57         items:
58           $ref:
59 ↪     "./00_common_definitions.yml#/definitions/item_requirement_definition"
60
61   actual_tool_usage:
62     description: identifies the tools used to perform the work.
63     oneOf:
64       - type: null
65       - type: array
66         items:
67           $ref:
68 ↪     "./00_common_definitions.yml#/definitions/tool_requirements_definition"
69
70   actual_services_usage:
71     description: identifies (contracted) service that was actually needed complete
72 ↪     the work.
73     oneOf:
74       - type: null
75       - type: array
76         items:
77           $ref:
78 ↪     "./00_common_definitions.yml#/definitions/service_requirement_definition"
79
80   actual_trades_involvement:
81     description: identifies the trade and qualifications needed to complete the
82 ↪     work.
83     oneOf:
84       - type: null
85       - type: array
86         items:
87           $ref:
88 ↪     "./00_common_definitions.yml#/definitions/trade_requirement_definition"
89
90 # CHILDREN WORK-ORDER DOCUMENTATION
91
92 # The user is able to account for additional work done, by adding new
93 ↪ children_work_documentation items. These items would refer any work order.  []
94 ↪ todo: need to define a procedure for creating a new work documentation.
95
96 children_work_documentations:
97   description:
98   oneOf:
```

```
91     - type: null
92     - type: array
93       items:
94         $ref: "../A-entity_record_schema/36_work_documentation.yml"
95
96 # FAILURE REPORTS AND FOLLOWUP REQUESTS
97
98 # Note: the failures are reported on follow-up work requests. The linkage between
↪ a followup work order and work order documentation is found on the work order
↪ schema
99
100 # COMMON RECORD PROPERTIES
101
102 duplicate_record_of:
103   $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
```

41_meter.yml

```
1 $schema: "http://json-schema.org/draft-07/schema#"
2 title: Maximo Meter
3 description: A JSON Schema representing a Meter object in IBM Maximo.
4 $id:
↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/1-Schemas/A-entity_record_schema/41_meter.y
5 type: object
6 properties:
7
8   Id:
9     type:
10       - string
11       - "null"
12     description: Unique identifier for the meter record.
13
14   description:
15     type:
16       - string
17       - "null"
18     description: optional description or notes about the meter.
19
20   asset:
21     oneOf:
22       - $ref: "../01_asset.yml"
23       - type: null
24     description: identifier of the asset associated with this meter.
25
26   functional-location:
27     oneOf:
28       - $ref: "../02_functional-location.yml"
29       - type: null
30     description: identifier of the functional-location associated with this meter.
31
32   meter_name:
```



```
33   type:
34     - string
35     - "null"
36   description: name of the meter.
37
38 meter_type:
39   type:
40     - string
41     - "null"
42   enum:
43     - HOURS
44     - CYCLES
45     - MILES
46     - KILOMETERS
47     - ...
48   description: type of meter measurement.
49   $comment: the enum values is not meant to be comprehensive
50
51 unit_of_measure:
52   type:
53     - string
54     - "null"
55   description: unit of measure for the meter (e.g., hours, cycles).
56
57 current_reading:
58   type:
59     - number
60     - "null"
61   description: is the current meter reading.
62
63 last_reading:
64   type:
65     - number
66     - "null"
67   description: is the latest know meter reading
68
69 # COMMON RECORD PROPERTIES
70
71 duplicate_record_of:
72   $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
73
74 record_retirement_information:
75   $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
```

Folder: B-entity_class_object_schema

00_common_class_definitions.yml

```
1 ---
2 $schema: "http://json-schema.org/draft-07/schema#"
3 title: generic class object
```

```
4 $id:
  ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/B-entity_class_object_schema
5 type: object
6
7 # This set of properties are used in all classification objects (e.g., asset,
  ↪ org-chart group, etc.)
8
9 properties:
10
11   class_name:
12     type: string
13     description: a noun or short noun-phrase name of the class
14
15   class_description:
16     oneOf:
17       - type: null
18       - type: string
19
20   synonym_names:
21     oneOf:
22       - type: null
23       - type: array
24         items:
25           type: string
26     description: other synonymous names that may be used by a user in search
27
28   can_be_applied_to_instances:
29     type: boolean
30     $comment: |
31       a "FALSE" value indicates that the class is meant to be a structural part of
  ↪ the classification tree, and cannot be used to classify any entity (i.e. asset,
  ↪ functional-location, physical-location, etc).
32
33   # COMMON RECORD PROPERTIES
34
35   duplicate_record_of:
36
37     $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
38
39   record_retirement_information:
40
41     $ref: "../00_common_definitions.yml#/definitions/record_retirement_def"
```

01_asset_item_tool_class.yml

```
1 ---
2 $schema: "http://json-schema.org/draft-07/schema#"
3 title: asset item class
4 $id:
  ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/B-entity_class_object_schema
5 type: object
```

```
6
7 #####
8 # 1. Properties
9 #####
10
11 properties:
12
13   allOf:
14     - $ref:
15       ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
16       #inherit the definitions and rules from th
17       ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
18
19   parent_class:
20     $ref: B-entity_class_object_schema/01_asset_item_tool_class.yml
21
22   only_used_as_a_part:
23     oneOf:
24       - type: boolean
25       - type: null
26     description: A true value indicates that all instances of this class is always
27       ↪ used as a part of another asset, and would never be given an asset tag. E.g.,
28       ↪ bearing.
29     rule_spec:
30       - name: Do not include only_used_as_a_part in the asset classification
31         spec_ID: NJ1E1ZbOgg
32         status: TBS
33       $comment: A false or null value materially mean the same thing.
```

02_functional-location_class.yml

```
1 ---
2 $schema: "http://json-schema.org/draft-07/schema#"
3 title: functional-location classification object
4 $id:
5   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/B-entity_class_object_schema
6 type: object
7
8 #####
9 # 1. Properties
10 #####
11
12 properties:
13
14   allOf:
15     - $ref:
16       ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
17       #inherit the definitions and rules from th
18       ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
```

```

16
17 parent_class:
18     $ref: B-entity_class_object_schema/02_functional-location_class.yml
19     sort order: 1-30
20
21 discrete_asset_functional-location:
22     oneOf: [type: boolean, type: null]
23     read-only: true
24     description: a functional-location that can be occupied by a single discrete
↵ asset.
25
26 functional_structure_functional-location:
27     oneOf: [type: boolean, type: null]
28     read-only: true
29     description: a functional-location that cannot be occupied any discrete asset,
↵ but can be the parent to other functional-locations.
30
31 defined_set_of_functional-locations:
32     oneOf: [type: boolean, type: null]
33     read-only: true
34     description: a functional-location that can be occupied by a collection of
↵ discrete assets.
35
36 compatible_asset_occupant_classes:
37     oneOf:
38         - type: array
39           items:
40             $ref: "./01_asset.yml"
41         - type: null
42     $comment: |
43         [ ]RULE NJQ6BwsVee: A asset must be an instance of one of classes listed in
↵ this field to be allowed to occupy a functional-location under this class.
44
45 #####
46 # RULES
47 #####
48
49 rule_spec:
50     - name: Is an functional-location, functional structure functional-location, or
↵ defined set of functional-locations
51       spec_ID: NyD4XGbuex
52       specification: |
53           if functional-locationClassX is a descendent of "Discrete Asset
↵ functional-location" in the functional-location classification hierarchy:
54               set functional-locationClassX.properties."a discrete asset
↵ functional-location" to TRUE
55               set functional-locationClassX.properties."a functional structure
↵ functional-location" to FALSE
56               set functional-locationClassX.properties."a defined ser of
↵ functional-location" to FALSE
57           elif functional-locationClassX is a descendent of "functional structure
↵ functional-location" in the functional-location classification hierarchy:
58               set functional-locationClassX.properties."a discrete asset
↵ functional-location" to FALSE

```

```
59         set functional-locationClassX.properties."a functional structure
↪ functional-location" to TRUE
60         set functional-locationClassX.properties."a defined ser of
↪ functional-location" to FALSE
61         elif functional-locationClassX is a descendent of "Defined Set of
↪ functional-locations" in the functional-location classification hierarchy:
62         set functional-locationClassX.properties."a discrete asset
↪ functional-location" to FALSE
63         set functional-locationClassX.properties."a functional structure
↪ functional-location" to FALSE
64         set functional-locationClassX.properties."a defined ser of
↪ functional-location" to TRUE
65         status: specified
```

03_physical_location_class.yml

```
1 title: functional-location classification object
2 $id:
↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/B-entity_class_object_schema
3 $schema: "http://json-schema.org/draft-07/schema#"
4 type: object
5
6 properties:
7
8     allOf:
9     - $ref:
↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
10     #inherit the definitions and rules from th
↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
11
12     parent_class:
13     $ref: B-entity_class_object_schema/03_space_class.yml
```

04_org-chart_group_class.yml

```
1 ---
2 $schema: "http://json-schema.org/draft-07/schema#"
3 title: functional-location classification object
4 $id:
↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/B-entity_class_object_schema
5 type: object
6
7 properties:
8
9     allOf:
10     - $ref:
↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
11     #inherit the definitions and rules from th
↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
```

```
12
13   parent_class:
14     $ref: B-entity_class_object_schema/04_org_class.yml
```

32_discrete_activity_class.yml

```
1  ---
2  title: discrete activity classification object
3  $id:
4    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/B-entity_class_object_schema
5  $schema: "http://json-schema.org/draft-07/schema#"
6  type: object
7  properties:
8
9    allOf:
10      - $ref:
11        ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
12        #inherit the definitions and rules from th
13        ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
14
15  parent_class:
16    $ref: B-entity_class_object_schema/32_discrete_activity_class.yml
```

33_work_type_class.yml

```
1  title: work type object
2  $id:
3    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/B-entity_class_object_schema
4  $schema: 'http://json-schema.org/draft-07/schema'
5  type: object
6  properties:
7
8    allOf:
9      - $ref:
10        ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
11        #inherit the definitions and rules from th
12        ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
13
14  parent_class:
15    $ref: B-entity_class_object_schema/33_work_type_class.yml
```

Folder: 2-Classification_Trees

01_asset_classification.md

```
1 > **Important Note:**
2 >
3 > **This section no longer being maintained**. The asset classification has been
   ↳ ported to an ontology file for further maintenance. You can find the file at
   ↳ https://github.com/TW-ASMP/TWONTO/blob/main/OWL/TWONTO.ttl
4 >
5 > To view the classification, save it to you computer, and open it with the desktop
   ↳ version of [Stanford Protege] (https://protege.stanford.edu/software.php).
6
```

02_functional-location_classification.md

```
1 ## Top-Level functional-location Classification
2
3 The following is a sample of the Toronto Water's functional-location classification.
   ↳
4
5 * Discrete Asset functional-location*
6     * generator functional-location
7         * backup generator functional-location
8         * emergency generator functional-location
9     * breaker functional-location
10        * bus feeder breaker functional-location
11        * line protection breaker functional-location
12        * load breaker functional-location
13        * tie-breaker functional-location
14 * Collection of Assets functional-location*
15     * functional structure functional-location
16         * system block
17     * facility
18         * pumping station
19         * water treatment facility
20         * wastewater treatment facility
21         * lab
22         * yard
23     * process
24 * Linear Functional Structure functional-location*
25     * system train functional-location
26     * line functional-location
27         * simple line functional-location
28         * primary path line functional-location
29     * junction functional-location
30 * Defined Set of functional-locations*
31     * defined set of discrete assets
32     * defined set of functional structures
33
```

```
34
35 ## Requirements for Implementer
36
37 * []REQ Nyh7RPjEgl #IMP "classes names specified in title-case and with an asterisk
  ↳ symbol shall have the applicable to individual property, found in the class
  ↳ object, set to false"
38
39 ## Notes
40
41 * []TODO #TW: the following should be moved to the asset classification.
42   * system on a skid
43   * system of standardized modular parts
```

03_physical-location_classification.md

```
1 ## Top-Level Space Classification and Examples
2
3 The following is a sample of the Toronto Water's physical-location classification.
4
5 * Discrete Space*
6   * building or structure interior
7   * room interior
8   * corridor
9   * mezzanine
10  * stairwell
11  * stairwell segment
12  * tunnel interior
13  * equipment cabinet interior
14  * storage cabinet interior
15  * facility site physical-location
16  * facility site section
17  * building or structure exterior
18  * vehicle interior
19 * Collection of Spaces*
20   * physical-locations of a building or structure [^1]
21   * define set of physical-locations
22
23 ## Requirements for Implementer
24
25 []REQ Nyh7RPjEgl #IMP "classes names specified in camel-case and with an asterisk
  ↳ symbol shall have the applicable to individual property, found in the class
  ↳ object, set to false"
26
27 ## Notes
28 ### Footnotes
29 [^1]: a collection of indoor and outdoor physical-locations inside and around a
  ↳ structure. []TODO #TW "add to the TWONTO".
```


04_org-chart_group_classification.md

```
1  ## Top-Level Org/Group Classification
2
3  The following is a sample of the Toronto Water's org-chart group classification.
4
5  * Level of Government*
6      * provincial government
7      * regional government
8      * municipal government
9  * Group in the City*
10     * cluster
11     * division
12     * section
13     * unit
14     * Group in TW*
15         * work area
16         * crew
17 * government agency
18 * private business
19 * non-governmental organization
20
21 ## Requirements for Implementer
22
23 []REQ Nyh7RPjEgl #IMP "classes names with an asterisk symbol shall have the
    ↳ applicable to individual property, found in the class object, set to false"
```

31_work_type.md

```
1  ## Work Type [^2]
2
3  The following is the complete set of Toronto Water's work type classification.
4
5  * corrective
6  * emergency [^1]
7  * investigative
8  * preventive
9  * informational
10 * project
11     * contractor support
12
13 ## Notes
14 [^1]: An emergency work order is technically a corrective work that must be done
    ↳ urgently; may also involve an investigative component (not unlike other
    ↳ corrective work orders).
15 [^2]: The commissioning work type has been removed from this list but retained in
    ↳ the discrete activity classification.
```

32_discrete_activity_classification.md

```
1  ## Top-Level Discrete Activity Classification
2
3  The following is is the complete set of Toronto Water's discrete activity
   ↳ classification.
4
5  * Condition Evaluation*
6    * quick check
7    * inspection and evaluation
8    * test and analysis
9    * condition analysis
10 * sample collection
11 * cause investigation
12 * repair or service [^1]
13   * calibration
14   * asset replacement
15   * asset part replacement
16   * asset part movement
17 * Move or Replace*
18   * new asset installation
19   * asset movement
20   * asset part movement
21   * asset replacement
22   * asset part replacement
23   * asset hand-over
24 * Life Cycle Events*
25   * asset commissioning
26   * asset hand-over [^3]
27   * final asset decommissioning
28 * Asset Modification*
29   * modify asset set-point
30   * physical modification to asset
31   * physical modification to building or structure [^2]
32 * asset assignment
33 * Contributory Work*
34   * item procurement
35   * work coordination
36   * safety preparation
37   * setup
38   * takedown of setup
39   * travel
40 * design or redesign
41   * creation of new functional-location
42   * removal of existing functional-location
43 * Asset Data*
44   * record information correction
45   * record retirement
46
47 ## Requirements for Implementer
48
49 ```yaml
```

```
50
51 rule_spec:
52   - name: Valid Assignment of an Asset
53     spec_ID: 01JDCNEFAED17CWF2K851ZAJKW
54     type: [assertion]
55     description: |
56       classes names specified with an asterisk symbol shall have the their
↵     .property.can_be_applied_to_instances value set to false
57
58   ~ ~ ~
59
60
61 ### Footnotes
62 [^1]: more will be added before the final implementation.
63 [^2]: the physical modification of a building or a structure may result in the
↵     creation and removal of a physical-location, hence it is singled out.
64 [^3]: the process by which a asset's ownership is transferred from a capital project
↵     to Toronto Water.
```

Folder: 3-System_Hierarchies

02_functional-location_hierarchy.md

```
1 # TW Highest Level Hierarchy
2
3 ## The Hierarchy
4 - TW System
5   - Drinking Water Network
6     - Drinking Water Treatment Plants [^1]
7     - Distribution Pumping Stations [^2]
8     - Storage Assets in Drinking Water Supply Network [^3]
9   - Waste and Storm Water Network
10     - Collection Pumping Stations [^4]
11     - Chambers in Sewer Network
12     - Storages of Wet Whether Flow [^5]
13     - Wastewater Treatment Plants [^6]
14   - Yards
15   - Independent Buildings
16
17 ## The Significance in the Usage of Plurals
18 Where a plural noun is used, for example: Water Treatment Plants, the entity
↵   represents a set of things. In the case of the example - the set of water
↵   treatment plants in TW.
19
20 ## Notes
21
22 [^1]: i.e. {FCL}, {FIS}, {FHO}, {FHA}
23 [^2]: the set of 18 pumping stations
24 [^3]: the set of all reservoirs and elevated tanks
25 [^4]: pumping station for waste and storm water
```

```
26  [^5]: the set of all wet-whether storage, inline an offline.  
27  [^6]: i.e. {THC}, {THR}, {TAB}, {TNT}
```

03_physical-location_hierarchy.md

```
1  # TW Highest Level Spatial Hierarchy  
2  
3  ## The Hierarchy  
4  - Spaces in TW  
5      - Spaces in Drinking Water System  
6          - Spaces in Drinking Water Treatment  
7          - Spaces in Distribution Pumping Stations  
8          - Spaces in Storage of Drinking Water  
9      - Spaces in Waste and Storm Water System  
10         - Spaces in Collection Pumping Stations  
11         - Spaces in Chambers in Sewer Network  
12         - Spaces in Storages of Wet Whether Flow  
13         - Spaces in Wastewater Treatment Plants  
14     - Spaces in Yards  
15     - Spaces in Independent Buildings  
16  
17  ## The Significance in the Usage of Plurals  
18  Where a plural noun is used, for example: Spaces in Drinking Water Treatment Plants,  
19  ↪ the entity represents a set of things. In the case of the example - the set  
20  ↪ physical-locations within the drinking water treatment plants in TW.  
  
## Notes
```

04_org-chart_group_hierarchy.md

```
1  # Organizations in TW and Interact with TW  
2  
3  ## The Hierarchy  
4  * Region of Durham  
5  * York Region  
6  * Peel Region  
7  * Province of Ontario  
8  * Metrolinx  
9  * TRCA  
10 * Envave Energy  
11 * City of Toronto  
12     * Solid Waste  
13     * Toronto Water  
14         * Distribution & Collection Section  
15             * DOS  
16             * WASP  
17             * Central Services  
18             * Program Maintenance
```

```
19      * Water Treatment & Supply Section
20      * Wastewater Treatment Section
21          * Ashbridge's Bay Wastewater Treatment Plant
22          * Highland Creek Wastewater Treatment Plant
23              * Work Area 1
24              * Work Area 2
25
26 ## Notes
27 * the portion of this hierarchy under TW is not complete; it will be completed
  ↪ before Phase 3 of implementation is complete.
```

Folder: 4-Class_Dependent_Specifications

Folder: A-asset_class_properties

01_pump.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: pump
4  $id:
  ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/01_pu
5  type: object
6
7  properties:
8
9      pump_type:
10         oneOf:
11             - type: null
12             - type: string
13         description: Indicates the pump type
14         enum:
15             - "dynamic, centrifugal"
16             - "dynamic, axial"
17             - "reciprocating, piston or plunger"
18             - "reciprocating, diaphragm"
19             - "rotary, vane"
20             - "rotary, piston"
21             - "screw pump"
22             - "gear pump"
23
24         orientation:
25             oneOf:
26                 - type: null
27                 - type: string
28             description: Indicates how the pump is oriented in 3D physical-location
29             enum:
30                 - "horizontal"
31                 - "vertical"
32                 - "angled"
```

```
33     - "inverted"
34
35 variable_speed:
36   oneOf:
37     - type: null
38     - type: boolean
39   description: Indicates the if the pump has variable speed control
40
41 max_RPM:
42   oneOf:
43     - type: null
44     - type: number
45   description: Indicates the maximum RPM for the pump
46
47 max_flow:
48   oneOf:
49     - type: null
50     - type: number
51   description: Indicates the maximum flow rate at the maximum RPM for the pump in
↪ L/s
52
53 pump_head:
54   oneOf:
55     - type: null
56     - type: number
57   description: Indicates the pressure head for the pump in metres
58
59 submersible:
60   oneOf:
61     - type: null
62     - type: boolean
63   description: Indicates if the pump is submersible
64
65 drive_coupling_type:
66   oneOf:
67     - type: null
68     - type: string
69   description: Indicates how the drive and pump are coupled together
70   enum:
71     - "direct drive"
72     - "belt drive"
73     - "gear drive"
74     - "flexible"
75     - "chain drive"
76     - "hydraulic"
77
78 drive_type:
79   oneOf:
80     - type: null
81     - type: string
82   description: Indicates what the mechanically drives the pump
83   enum:
84     - "electric motor"
```

```
85     - "engine"
86
87   bearings_are_sealed:
88     oneOf:
89       - type: null
90       - type: boolean
91     description: Indicates if the pump has sealed bearings
92
```

02_motor.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: motor
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/02_motor.yml
6
7  $comment: >
8
9  properties:
10
11    type:
12      oneOf:
13        - type: null
14        - type: string
15      description: Indicates the motor type
16      enum:
17        - "AC"
18        - "AC, squirrel cage induction"
19        - "AC, wound rotor induction"
20        - "AC, synchronous"
21        - "DC"
22        - "DC, separately excited"
23        - "DC, self excited"
24        - "DC, permanent magnet"
25
26    voltage:
27      oneOf:
28        - type: null
29        - type: number
30      description: Indicates the voltage of the motor in Volts
31
32    horse_power:
33      oneOf:
34        - type: null
35        - type: number
36      description: Indicates the horse power of the motor
37
38    NEMA_frame_type:
39      oneOf:
```

```
40     - type: null
41     - type: string
42   description: Indicates the NEMA frame type for the motor
43   enum:
44     - "42"
45     - "48"
46     - "56"
47     - "66"
48     - "182"
49     - "184"
50     - "213"
51     - "215"
52     - "1412AT"
53     - "143T"
54     - "145T"
55     - "146AT"
56     - "148AT"
57     - "149AT"
58     - "182AT"
59     - "182T"
60     - "184T"
61     - "186ACY"
62     - "186AT"
63     - "189AT"
64     - "203#"
65     - "204#"
66     - "2110AT"
67     - "213T"
68     - "215T"
69     - "219AT"
70     - "224#"
71     - "225#"
72     - "254#"
73     - "254T"
74     - "254U"
75     - "256T"
76     - "256U"
77     - "284#"
78     - "284T"
79     - "284TS"
80     - "284U"
81     - "286T"
82     - "286TS"
83     - "286U"
84     - "324#"
85     - "324T"
86     - "324TS"
87     - "324U"
88     - "326#"
89     - "326T"
90     - "326TS"
91     - "326U"
92     - "364#"
```



```
93     - "364S#"
94     - "364T"
95     - "364TS"
96     - "364U"
97     - "365#"
98     - "365T"
99     - "365TS"
100    - "365U"
101    - "404T"
102    - "404TS"
103    - "404U"
104    - "405T"
105    - "405TS"
106    - "405U"
107    - "444T"
108    - "444TS"
109    - "444U"
110    - "445T"
111    - "445TS"
112    - "445U"
113    - "447T&&"
114    - "447TS&&"
115    - "449T"
116    - "449TS"
117    - "48H"
118    - "56H"
119    - "56HZ"
120    - "L182ACY"
121    - "L186AT"
122
123  NEMA_enclosure_type:
124    oneOf:
125      - type: null
126      - type: string
127    description: Indicates what the NEMA enclosure type for the motor
128    enum:
129      - ODP
130      - TEFC
131      - TENV
132      - TEAO
133      - TEWD
134      - EXPL
135      - HAZ
136
137  bearings_are_sealed:
138    oneOf:
139      - type: null
140      - type: boolean
141    description: Indicates if the motor has sealed bearings
142
```

03_valve.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: valve
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/03_valve.yml
6  type: object
7  $comment: >
8
9  properties:
10
11    type:
12      oneOf:
13        - type: null
14        - type: string
15      description: Indicates the valve type
16      enum:
17        - "ball"
18        - "butterfly"
19        - "cone"
20        - "diaphragm"
21        - "gate valve"
22        - "globe valve"
23        - "knife valve"
24        - "needle valve"
25        - "pinch valve"
26        - "plug valve"
27      $comment: may be redundant to label them valve in the valve type
28
29    special_type:
30      oneOf:
31        - type: null
32        - type: string
33      description: Indicates what type of specialized valve it is if it is a
34      ↪ specialized valve
35      enum:
36        - "air release valve"
37        - "backflow preventer"
38        - "check"
39        - "pressure relief valve"
40        - "solenoid"
41      $comment: may be redundant to label them valve in the valve type
42
43    size:
44      oneOf:
45        - type: null
46        - type: number
47      description: Indicates the valve size in inches
48
49    ANSI_type:
```

```
49   oneOf:
50     - type: null
51     - type: string
52   description: Indicates the ANSI type for the valve
53   enum:
54     - 150
55     - 300
56     - 400
57     - 600
58     - 900
59     - 1500
60     - 2500
61     - 4500
62
63   ANSI_class:
64     oneOf:
65       - type: null
66       - type: string
67     description: Indicates the ANSI class for the valve
68     enum:
69       - "A - Standard"
70       - "B - Special"
71       - "Limited"
72
73   horse_power:
74     oneOf:
75       - type: null
76       - type: number
77     description: Indicates the horse power of the motor
78
79   NEMA_frame:
80     oneOf:
81       - type: null
82       - type: string
83     description: Indicates the NEMA frame type for the motor
84     enum:
85       - "42"
86       - "48"
87       - "56"
88       - "66"
89       - "182"
90       - "184"
91       - "213"
92       - "215"
93       - "1412AT"
94       - "143T"
95       - "145T"
96       - "146AT"
97       - "148AT"
98       - "149AT"
99       - "182AT"
100      - "182T"
101      - "184T"
```

```
102     - "186ACY"
103     - "186AT"
104     - "189AT"
105     - "203#"
106     - "204#"
107     - "2110AT"
108     - "213T"
109     - "215T"
110     - "219AT"
111     - "224#"
112     - "225#"
113     - "254#"
114     - "254T"
115     - "254U"
116     - "256T"
117     - "256U"
118     - "284#"
119     - "284T"
120     - "284TS"
121     - "284U"
122     - "286T"
123     - "286TS"
124     - "286U"
125     - "324#"
126     - "324T"
127     - "324TS"
128     - "324U"
129     - "326#"
130     - "326T"
131     - "326TS"
132     - "326U"
133     - "364#"
134     - "364S#"
135     - "364T"
136     - "364TS"
137     - "364U"
138     - "365#"
139     - "365T"
140     - "365TS"
141     - "365U"
142     - "404T"
143     - "404TS"
144     - "404U"
145     - "405T"
146     - "405TS"
147     - "405U"
148     - "444T"
149     - "444TS"
150     - "444U"
151     - "445T"
152     - "445TS"
153     - "445U"
154     - "447T&&"
```

```
155     - "447TS&&"
156     - "449T"
157     - "449TS"
158     - "48H"
159     - "56H"
160     - "56HZ"
161     - "L182ACY"
162     - "L186AT"
163
164 NEMA_enclosure_type:
165   oneOf:
166     - type: null
167     - type: string
168   description: Indicates what the NEMA enclosure type for the motor
169   enum:
170     - ODP
171     - TEFC
172     - TENV
173     - TEO
174     - TEWD
175     - EXPL
176     - HAZ
177
178 bearings_are_sealed:
179   oneOf:
180     - type: null
181     - type: boolean
182   description: Indicates if the motor has sealed bearings
183
184 cold_working_pressure:
185   oneOf:
186     - type: null
187     - type: number
188   description: Indicates the cold working pressure of the valve in psi
189
190 nominal_pressure:
191   oneOf:
192     - type: null
193     - type: number
194   description: Indicates the nominal pressure of the valve in psi
195
196 valve_body_type:
197   oneOf:
198     - type: null
199     - type: string
200   description: Indicates the valve body material
201   enum:
202     - "carbon steel"
203     - "stainless steel"
204     - "duplex"
205     - "alloy"
206     - "composite"
207     - "titanium"
```

```
208
209 actuator_type:
210   oneOf:
211     - type: null
212     - type: string
213   description: Indicates the type of actuator
214   enum:
215     - "electric"
216     - "pneumatic"
217     - "hydraulic"
218     - "manual"
219
220 stem_seal_type:
221   oneOf:
222     - type: null
223     - type: string
224   description: Indicates the type of stem seal for the valve
225   enum:
226     - "duplex"
227     - "lip seal"
228     - "o-ring"
229     - "stuffing box"
230
231 valve_turn_direction:
232   oneOf:
233     - type: null
234     - type: string
235   enum:
236     - "counter-clockwise"
237     - "clockwise"
238   $comment: This specification is specifically requested by DOS via Silvia Sawada
239
240
```

04_breaker.yml

```
1 ---
2 $schema: "http://json-schema.org/draft-07/schema#"
3 title: breaker
4 $id:
5   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/04_br
6 type: object
7 $comment: >
8
9 properties:
10
11   breaker_type:
12     oneOf:
13       - type: null
14       - type: string
```

```
15     description: Indicates the breaker type
16     enum:
17       - "insulated case"
18       - "metal clad or enclosed"
19       - "molded case"
20
21     max_voltage:
22       oneOf:
23         - type: null
24         - type: number
25       description: Indicates what the maximum continuous voltage rating for the
↪ breaker in Volts
26
27     max_amperage:
28       oneOf:
29         - type: null
30         - type: number
31       description: Indicates what the maximum continuous current rating for the
↪ breaker in Amps
32
33     main_contactor_type:
34       oneOf:
35         - type: null
36         - type: string
37       description: Indicates the ANSI type for the valve
38       enum:
39         - "air insulated"
40         - "air insulated, air blast"
41         - "vacuum insulated"
42         - "oil insulated"
43         - "gas insulated"
44
45
```

05_starter.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: starter
4  $id:
↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/05_starter.yml
5  type: object
6
7  properties:
8
9    voltage_rating:
10      oneOf:
11        - type: null
12        - type: number
13      description: Indicates the continuous voltage rating for the starter in Volts
14
```

```
15   current_rating:
16     oneOf:
17       - type: null
18       - type: number
19     description: Indicates the continuous current rating for the starter in Amps
20
21   has_VFD_function:
22     oneOf:
23       - type: null
24       - type: boolean
25     description: Indicates the presence of a variable frequency drive within the
26 ↪ starter
27
28   has_soft_starting_function:
29     oneOf:
30       - type: null
31       - type: boolean
32     description: Indicates the function of a soft starter in the starter
33
34   main_contactor_type:
35     oneOf:
36       - type: null
37       - type: string
38     description: Indicates the ANSI type for the valve
39     enum:
40       - "air insulated"
41       - "air insulated, air blast"
42       - "vacuum insulated"
43       - "oil insulated"
44       - "gas insulated"
```

06_transformer.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: transformer
4  $id:
5  ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/06_tr
6  type: object
7  $comment: >
8
9  properties:
10
11    primary_voltage:
12      oneOf:
13        - type: null
14        - type: number
15      description: Indicates the primary voltage rating for the transformer in Volts
16
```



```
17 secondary_voltage:
18   oneOf:
19     - type: null
20     - type: number
21   description: Indicates the secondary voltage rating for the transformer in Volts
22
23 power_rating:
24   oneOf:
25     - type: null
26     - type: number
27   description: Indicates the power rating for the transformer in kVA
28
29 oil_filled:
30   oneOf:
31     - type: null
32     - type: boolean
33   description: Indicates the requirement for oil cooling for the transformer
34
35 pressure_relay:
36   oneOf:
37     - type: null
38     - type: boolean
39   description: Indicates the presence of a sudden pressure relay
40
41 cooling_air_fan:
42   oneOf:
43     - type: null
44     - type: boolean
45   description: Indicates the presence of a cooling air fan(s)
46
47 coolant_pump:
48   oneOf:
49     - type: null
50     - type: boolean
51   description: Indicates the presence of a coolant pump
52
53 gas_monitor:
54   oneOf:
55     - type: null
56     - type: boolean
57   description: Indicates the presence of a gas monitor
58
59
```

07_HVAC.yml

```
1 ---
2 $schema: "http://json-schema.org/draft-07/schema#"
3 title: HVAC
4 $id:
5   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/07_HVAC
```

```

5  type: object
6
7  $comment: >
8
9  properties:
10
11    CFM_rating:
12      oneOf:
13        - type: null
14        - type: number
15      description: Indicates the capacity rating for the HVAC unit in CFM
16
17    static_pressure:
18      oneOf:
19        - type: null
20        - type: number
21      description: Indicates the static pressure for the HVAC unit in inch water
22      ↪ column
23
24    has_heating_function:
25      oneOf:
26        - type: null
27        - type: boolean
28      description: Indicates the presence of a heating function in the HVAC unit such
29      ↪ as heating coil or gas burner
30
31    has_cooling_function:
32      oneOf:
33        - type: null
34        - type: boolean
35      description: Indicates the presence of a cooling function in the HVAC unit such
36      ↪ as cooling coil
37
38    has_dehumidification_function:
39      oneOf:
40        - type: null
41        - type: boolean
42      description: Indicates the presence of a dehumidifier
43
44    has_maintainable_damper:
45      oneOf:
46        - type: null
47        - type: boolean
48      description: Indicates the presence of a maintainable damper
49      $comment:
50
51    uses_belt:
52      oneOf:
53        - type: null
54        - type: boolean
55      description: Indicates the presence of replaceable belts

```

55

56

08_blower_fan.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: blower_fan
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/08_bl
6  type: object
7  $comment: >
8
9  properties:
10
11    CFM_rating:
12      oneOf:
13        - type: null
14        - type: number
15      description: Indicates the capacity rating for the HVAC unit in CFM
16
17    static_pressure:
18      oneOf:
19        - type: null
20        - type: number
21      description: Indicates the static pressure for the HVAC unit in inch water
22    ↪ column
23
24    drive_coupling_type:
25      oneOf:
26        - type: null
27        - type: string
28      description: Indicates the type of drive coupling
29      enum:
30        - "direct drive"
31        - "belt drive"
32        - "gear drive"
33        - "flexible"
34        - "chain drive"
35        - "hydraulic"
36
37    bearings_are_sealed:
38      oneOf:
39        - type: null
40        - type: boolean
41      description: Indicates the presence of sealed of shielded bearings
42
```

09_compressor.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: compressor
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/09_co
6
7  type: object
8
9  $comment: >
10
11  properties:
12    rated_pressure:
13      oneOf:
14        - type: null
15        - type: number
16      description: Indicates the capacity rating for the compressor in kPa
17
18    rated_flow:
19      oneOf:
20        - type: null
21        - type: number
22      description: Indicates the capacity rating for the compressor in SCMH
23
24    drive_coupling:
25      oneOf:
26        - type: null
27        - type: string
28      description: Indicates the type of drive coupling
29      enum:
30        - "direct drive"
31        - "belt drive"
32        - "gear drive"
33        - "flexible"
34        - "chain drive"
35        - "hydraulic"
36
37    bearings_are_sealed:
38      oneOf:
39        - type: null
40        - type: boolean
41      description: Indicates the presence of sealed of shielded bearings
```

10_generator.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: generator
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/10_ge
```

```
5 type: object
6
7 properties:
8
9   fire_life-safety_elevator:
10     oneOf:
11       - type: null
12       - type: boolean
13     description: Indicates if the device directly supports any part of the fire
    ↪ suppression or any life safety systems
14
15   is_mobile:
16     oneOf:
17       - type: null
18       - type: boolean
19     description: Indicates if the generator is mobile
20
21   power_rating:
22     oneOf:
23       - type: null
24       - type: number
25     description: Indicates the power rating in kilo watts
26
27   voltage_rating:
28     oneOf:
29       - type: null
30       - type: number
31     description: Indicates the capacity rating for the generator in Volts
32
33   drive_type:
34     oneOf:
35       - type: null
36       - type: string
37     description: Indicates the type of drive
38     enum:
39       - "engine, diesel or bio-diesel"
40       - "engine, natural gas"
41       - "turbine"
42
43   is_brushless:
44     oneOf:
45       - type: null
46       - type: boolean
47     description: Indicates the presence of brushes within the generator
48
49   has_test_load:
50     oneOf:
51       - type: null
52       - type: boolean
53     description: Indicates the presence of an electrical connection to attach a test
    ↪ load
54
55   drive_coupling_type:
```

```
56     oneOf:
57       - type: null
58       - type: string
59     description: Indicates the type of drive coupling
60     enum:
61       - "direct drive"
62       - "belt drive"
63       - "gear drive"
64       - "flexible"
65       - "chain drive"
66       - "hydraulic"
67
68     bearings_are_sealed:
69       oneOf:
70         - type: null
71         - type: boolean
72     description: Indicates the presence of sealed or shielded bearings
```

11_UPS.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: ups
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/11_UPS.yml
6  type: object
7  $comment: >
8
9  properties:
10
11    powers_safety_load:
12      oneOf:
13        - type: null
14        - type: boolean
15      description: Indicates if the device directly supports any part of the fire
16      ↪ suppression system, life safety system, or elevator of a high building systems
17
18    battery_capacity:
19      oneOf:
20        - type: null
21        - type: number
22      description: Indicates the power rating in kilo watt hours
23
24    voltage_rating:
25      oneOf:
26        - type: null
27        - type: number
28      description: Indicates the output voltage of the UPS in Volts
29
30    battery_type:
```

```
30   oneOf:
31     - type: null
32     - type: string
33   description: Indicates the type of drive
34   enum:
35     - "sealed / valve regulated lead acid"
36     - "flooded / vented lead acid"
37     - "NiCad"
38     - "Li-ion"
39
40   has_integrated_charger:
41     oneOf:
42       - type: null
43       - type: boolean
44   description: Indicates that the UPS is physically integrated with charger
45
```

12_boiler.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: boiler
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/12_boiler.yml
6  type: object
7  $comment: >
8
9  properties:
10
11    energy_source:
12      oneOf:
13        - type: null
14        - type: string
15      description: Indicates the fuel source for the boiler
16      enum:
17        - "natural gas"
18        - "biogas / digester gas"
19        - "propane"
20        - "electric"
21        - "fuel oil"
22
23    boiled_medium:
24      oneOf:
25        - type: null
26        - type: string
27      description: Indicates the boiled/heated medium within the boiler
28      enum:
29        - water
30        - steam
31        - glycol
```

```
32
33 heat_rate:
34   oneOf:
35     - type: null
36     - type: number
37   description: Indicates the rated heat rate in the boiler in kilo Watts
38
39 max_pressure:
40   oneOf:
41     - type: null
42     - type: number
43   description: Indicates the maximum operating pressure for the boiler in psi
44
45 max_temperature:
46   oneOf:
47     - type: null
48     - type: number
49   description: Indicates the maximum operating temperature for the boiler in
↪ degrees celsius
50
51 heated_surface:
52   oneOf:
53     - type: null
54     - type: number
55   description: Indicates the heating surface area of the boiler in meters squared
56
57 capacity:
58   oneOf:
59     - type: null
60     - type: number
61   description: Indicates the size capacity of the boiler in Litres
62
63 TSSA_CRN:
64   oneOf:
65     - type: null
66     - type: string
67   description: Indicates the CRN number issued by the TSSA
```

13_pressure_vessel.yml

```
1 ---
2 $schema: "http://json-schema.org/draft-07/schema#"
3 title: pressure_vessel
4 $id:
↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/13_pr
5 type: object
6
7 $comment: >
8
9 properties:
10
```



```
11 contained_medium:
12   oneOf:
13     - type: null
14     - type: string
15   description: Indicates the medium within the pressure vessel
16   enum:
17     - "water"
18     - "steam"
19     - "glycol"
20     - "refrigerant"
21     - "compressed air"
22     - "digester gas"
23     - "ozone"
24
25 capacity:
26   oneOf:
27     - type: null
28     - type: number
29   description: Indicates the size capacity of the pressure vessel in Litres
30
31 max_pressure:
32   oneOf:
33     - type: null
34     - type: number
35   description: Indicates the maximum operating pressure for the pressure vessel in
36   ↪ psi
37
38 TSSA_CRN:
39   oneOf:
40     - type: null
41     - type: string
42   description: Indicates the CRN number issued by the TSSA
```

14_pressure_piping.yml

```
1 ---
2 $schema: "http://json-schema.org/draft-07/schema#"
3 title: pressure_piping
4 $id:
5   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/14_pressure_piping.yml
6 type: object
7 $comment: >
8
9 properties:
10
11   max_pipe_size_in_inches:
12     oneOf:
13       - type: null
14       - type: number
15     description: Indicates the maximum pipe size in inches
```

```
16
17 max_pressure:
18   oneOf:
19     - type: null
20     - type: number
21   description: Indicates the maximum working pressure within the pressure piping
↵ in kPa
22
23 max_temperature:
24   oneOf:
25     - type: null
26     - type: number
27   description: Indicates the maximum medium temperature within the pressure piping
↵ in degrees celsius
28
29 contained_medium:
30   oneOf:
31     - type: null
32     - type: string
33   description: Indicates the medium within the pressure piping
34   enum:
35     - "water"
36     - "steam"
37     - "glycol"
38     - "refrigerant"
39     - "compressed air"
40     - "digester gas"
41     - "ozone"
42
43 special_application:
44   oneOf:
45     - type: null
46     - type: string
47   description: Indicates the special application required for pressure piping
48   enum:
49     - "piping in fire protection system"
50     - "piping in heating system"
51     - "piping in refrigeration system"
52     - "compressed air piping"
53     - "hot oil piping"
54     - "buried water piping"
55
56 TSSA_CRN:
57   oneOf:
58     - type: null
59     - type: string
60   description: Indicates the CRN number issued by the TSSA
```

15_instrumentation.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: instrumentation
4  $id:
5    ↪ "https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/15_i
6  type: object
7  properties:
8
9    parameter:
10      oneOf:
11        - type: null
12        - type: string
13      description: Indicates the parameter that is being measured
14      enum:
15        - "density"
16        - "flow rate"
17        - "humidity"
18        - "level"
19        - "current"
20        - "power"
21        - "position"
22        - "pressure"
23        - "speed"
24        - "temperature"
25        - "torque"
26        - "uv"
27        - "vibration"
28        - "weight"
29        - "specific gravity"
30        - "ammonia"
31        - "carbon monoxide"
32        - "chlorination"
33        - "chlorine"
34        - "dissolved oxygen"
35        - "fluoride"
36        - "methane/lel"
37        - "total hydrocarbon"
38        - "orp"
39        - "ozone"
40        - "particulate"
41        - "ph"
42        - "sulphide"
43        - "sulphur dioxide"
44        - "suspended solids"
45        - "turbidity"
46
47    UOM:
48      oneOf:
49        - type: null
```

```
50     - type: string
51     description: Indicates the unit of measurement that the instrumentation is
↪ reporting values in
52
53     lower_bound:
54       oneOf:
55         - type: null
56         - type: number
57       description: Indicates the lower bound for the parameter of the instrument in
↪ the UOM
58
59     upper_bound:
60       oneOf:
61         - type: null
62         - type: number
63       description: Indicates the upper bound for the parameter of the instrument in
↪ the UOM
64
65     allOf:
66       - if:
67         properties:
68           parameter:
69             const: "Density"
70         then:
71         properties:
72           uom:
73             enum:
74               - "Kilograms Per Cubic Meter (kg/m³)"
75               - "Grams Per Cubic Centimeter (g/cm³)"
76               - "Grams Per Milliliter (g/mL)"
77               - "Pounds Per Cubic Foot (lb/ft³)"
78
79       - if:
80         properties:
81           parameter:
82             const: "Flow Rate"
83         then:
84         properties:
85           uom:
86             enum:
87               - "Liters Per Second (L/s)"
88               - "Cubic Meters Per Second (m³/s)"
89               - "Gallons Per Minute (GPM)"
90               - "Cubic Feet Per Minute (CFM)"
91               - "Liters Per Hour (L/h)"
92               - "Standard Cubic Feet Per Minute (SCFM)"
93
94       - if:
95         properties:
96           parameter:
97             const: "Humidity"
98         then:
99         properties:
```

```

100     uom:
101       enum:
102         - "Percentage (%)"
103         - "Grams Per Milliliter (g/mL)"
104         - "Milligrams Per Liter (mg/L)"
105         - "Parts Per Million (ppm)"
106         - "Parts Per Billion (ppb)"
107
108   - if:
109     properties:
110       parameter:
111         const: "Level"
112     then:
113       properties:
114         uom:
115           enum:
116             - "Centimeters (cm)"
117             - "Meters (m)"
118             - "Inch (in)"
119             - "Percentage (%)"
120             - "Feet (ft)"
121
122   - if:
123     properties:
124       parameter:
125         const: "Current"
126     then:
127       properties:
128         uom:
129           enum:
130             - "Ampere (A)"
131             - "Milliampere (mA)"
132
133   - if:
134     properties:
135       parameter:
136         const: "Power"
137     then:
138       properties:
139         uom:
140           enum:
141             - "Watt (W)"
142             - "kilowatt (kW)"
143             - "Megawatt (MW)"
144
145   - if:
146     properties:
147       parameter:
148         const: "Position"
149     then:
150       properties:
151         uom:
152           enum:

```

```
153         - "Centimeters (cm)"
154         - "Meters (m)"
155         - "Inch (in)"
156         - "Millimeter (mm)"
157         - "Feet (ft)"
158         - "Degree (°)"
159         - "Radians (rad)"
160         - "Unitless"
161
162     - if:
163         properties:
164             parameter:
165                 const: "Pressure"
166     then:
167         properties:
168             uom:
169                 enum:
170                     - "Pascal (Pa)"
171                     - "Kilopascal (kPa)"
172                     - "Bar"
173                     - "Atmosphere (atm)"
174                     - "Pounds Per Square Inch (PSI)"
175                     - "Millimeter of Mercury (mmHg)"
176                     - "Millimeter of Water (mmH2O)"
177                     - "Inch of Water (\\"WC)"
178
179     - if:
180         properties:
181             parameter:
182                 const: "Speed"
183     then:
184         properties:
185             uom:
186                 enum:
187                     - "Meters Per Second (m/s)"
188                     - "Kilometers Per Hour (km/h)"
189                     - "Feet Per Second (ft/s)"
190                     - "Mile Per Hour (mph)"
191                     - "Revolutions Per Minute (RPM)"
192
193     - if:
194         properties:
195             parameter:
196                 const: "Temperature"
197     then:
198         properties:
199             uom:
200                 enum:
201                     - "Degree Celsius (°C)"
202                     - "Degree Fahrenheit (°F)"
203
204     - if:
205         properties:
```

```

206     parameter:
207       const: "Torque"
208   then:
209     properties:
210       uom:
211         enum:
212           - "Newton-meters (N·m)"
213           - "Foot-pounds (ft·lb)"
214
215   - if:
216     properties:
217       parameter:
218         const: "UV"
219     then:
220       properties:
221         uom:
222           enum:
223             - "Watts Per Square Meter (W/m²)"
224             - "Percentage (%)"
225
226   - if:
227     properties:
228       parameter:
229         const: "Vibration"
230     then:
231       properties:
232         uom:
233           enum:
234             - "Meters Per Second (m/s)"
235             - "Centimeters Per Second (cm/s)"
236             - "Feet Per Second (ft/s)"
237             - "Inch Per Second (in/s)"
238             - "Meters Per Second Square (m/s²)"
239             - "Centimeters Per Second Square (cm/s²)"
240             - "Feet Per Second Square (ft/s²)"
241             - "Inch Per Second Square (in/s²)"
242             - "Hertz (Hz)"
243
244   - if:
245     properties:
246       parameter:
247         const: "Weight"
248     then:
249       properties:
250         uom:
251           enum:
252             - "Grams (g)"
253             - "Kilograms (kg)"
254             - "Pounds (lb)"
255             - "Metric Tons (tonne)"
256
257   - if:
258     properties:

```

```
259     parameter:
260       const: "Specific Gravity"
261   then:
262     properties:
263       uom:
264         enum:
265           - "Unitless"
266
267 - if:
268   properties:
269     parameter:
270       const: "Ammonia"
271   then:
272     properties:
273       uom:
274         enum:
275           - "Parts Per Million (ppm)"
276           - "Parts Per Billion (ppb)"
277           - "Milligrams Per Cubic Meter (mg/m³)"
278           - "Percentage (%)"
279           - "Micrograms Per Cubic Meter (µg/m³)"
280
281 - if:
282   properties:
283     parameter:
284       const: "Carbon Monoxide"
285   then:
286     properties:
287       uom:
288         enum:
289           - "Parts Per Million (ppm)"
290           - "Parts Per Billion (ppb)"
291           - "Milligrams Per Cubic Meter (mg/m³)"
292           - "Percentage (%)"
293           - "Micrograms Per Cubic Meter (µg/m³)"
294
295 - if:
296   properties:
297     parameter:
298       const: "Chlorination"
299   then:
300     properties:
301       uom:
302         enum:
303           - "Grams Per Milliliter (g/mL)"
304           - "Milligrams Per Liter (mg/L)"
305           - "Parts Per Million (ppm)"
306           - "Parts Per Billion (ppb)"
307
308 - if:
309   properties:
310     parameter:
311       const: "Chlorine"
```



```
312   then:
313     properties:
314       uom:
315         enum:
316           - "Grams Per Milliliter (g/mL)"
317           - "Milligrams Per Liter (mg/L)"
318           - "Parts Per Million (ppm)"
319           - "Parts Per Billion (ppb)"
320
321 - if:
322   properties:
323     parameter:
324       const: "Dissolved Oxygen"
325   then:
326     properties:
327       uom:
328         enum:
329           - "Grams Per Milliliter (g/mL)"
330           - "Milligrams Per Liter (mg/L)"
331           - "Parts Per Million (ppm)"
332           - "Parts Per Billion (ppb)"
333
334 - if:
335   properties:
336     parameter:
337       const: "Fluoride"
338   then:
339     properties:
340       uom:
341         enum:
342           - "Grams Per Milliliter (g/mL)"
343           - "Milligrams Per Liter (mg/L)"
344           - "Parts Per Million (ppm)"
345           - "Parts Per Billion (ppb)"
346
347 - if:
348   properties:
349     parameter:
350       const: "Methane/LEL"
351   then:
352     properties:
353       uom:
354         enum:
355           - "Parts Per Million (ppm)"
356           - "Parts Per Billion (ppb)"
357           - "Milligrams Per Cubic Meter (mg/m³)"
358           - "Percentage (%)"
359           - "Micrograms Per Cubic Meter (µg/m³)"
360
361 - if:
362   properties:
363     parameter:
364       const: "Total Hydrocarbon"
```

```
365   then:
366     properties:
367       uom:
368         enum:
369           - "Grams Per Milliliter (g/mL)"
370           - "Milligrams Per Liter (mg/L)"
371           - "Parts Per Million (ppm)"
372           - "Parts Per Billion (ppb)"
373
374 - if:
375   properties:
376     parameter:
377       const: "ORP"
378   then:
379     properties:
380       uom:
381         enum:
382           - "Grams Per Milliliter (g/mL)"
383           - "Milligrams Per Liter (mg/L)"
384           - "Parts Per Million (ppm)"
385           - "Parts Per Billion (ppb)"
386
387 - if:
388   properties:
389     parameter:
390       const: "Ozone"
391   then:
392     properties:
393       uom:
394         enum:
395           - "Grams Per Milliliter (g/mL)"
396           - "Milligrams Per Liter (mg/L)"
397           - "Parts Per Million (ppm)"
398           - "Parts Per Billion (ppb)"
399
400 - if:
401   properties:
402     parameter:
403       const: "Particulate"
404   then:
405     properties:
406       uom:
407         enum:
408           - "Grams Per Milliliter (g/mL)"
409           - "Milligrams Per Liter (mg/L)"
410           - "Parts Per Million (ppm)"
411           - "Parts Per Billion (ppb)"
412
413 - if:
414   properties:
415     parameter:
416       const: "PH"
417   then:
```

```
418     properties:
419       uom:
420         enum:
421           - "Unitless"
422
423   - if:
424     properties:
425       parameter:
426         const: "Sulphide"
427     then:
428       properties:
429         uom:
430           enum:
431             - "Grams Per Milliliter (g/mL)"
432             - "Milligrams Per Liter (mg/L)"
433             - "Parts Per Million (ppm)"
434             - "Parts Per Billion (ppb)"
435
436   - if:
437     properties:
438       parameter:
439         const: "Sulphur Dioxide"
440     then:
441       properties:
442         uom:
443           enum:
444             - "Parts Per Million (ppm)"
445             - "Parts Per Billion (ppb)"
446             - "Milligrams Per Cubic Meter (mg/m3)"
447             - "Percentage (%)"
448             - "Micrograms Per Cubic Meter (µg/m3)"
449
450   - if:
451     properties:
452       parameter:
453         const: "Suspended Solids"
454     then:
455       properties:
456         uom:
457           enum:
458             - "Grams Per Milliliter (g/mL)"
459             - "Milligrams Per Liter (mg/L)"
460             - "Parts Per Million (ppm)"
461             - "Parts Per Billion (ppb)"
462
463   - if:
464     properties:
465       parameter:
466         const: "Turbidity"
467     then:
468       properties:
469         uom:
470           enum:
```

```
471         - "Grams Per Milliliter (g/mL)"
472         - "Milligrams Per Liter (mg/L)"
473         - "Parts Per Million (ppm)"
474         - "Parts Per Billion (ppb)"
```

Folder: B-role_class_properties

01_pump_role.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: pump functional-location
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/4-Class_Dependent_Specifications/B-function
6  type: object
7  properties:
8
9    requires_variable_speed:
10      oneOf:
11        - type: null
12        - type: boolean
13      description: indicates the the pump serving in the functional-location must
14
15    required_max_flow:
16      oneOf:
17        - type: null
18        - type: number
19      description: indicates the maximum flow rate required to serve the
20    ↪ functional-location
21
22    required_total_dynamic_head:
23      oneOf:
24        - type: null
25        - type: number
26      description: Indicates the pressure head for the pump in metres
27
28    must_be_submersible:
29      oneOf:
30        - type: null
31        - type: boolean
32      description: Indicates if the pump is submersible
```

Folder: 5-Functions