

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	9
02_role.yml	21
03_space.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	44
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_role_class.yml	75
03_space_class.yml	77
04_org_class.yml	77
32_discrete_activity_class.yml	77
33_work_type.yml	78
Folder: 2-Classification_Trees	78
01_asset_classification.md	78
02_role_classification.md	78
03_space_classification.md	79
04_org_classification.md	80
31_work_type.md	81
32_discrete_activity_classification.md	81
Folder: 3-System_Hierarchies	82
02_role_hierarchy.md	82
03_space_hierarchy.md	83
04_org_hierarchy.md	84

Folder: 4-Class_Dependent_Specifications	84
Folder: A-asset_class_properties	84
01_pump.yml	84
02_motor.yml	86
03_valve.yml	89
04_breaker.yml	94
05_starter.yml	95
06_transformer.yml	96
07_HVAC.yml	97
08_blower_fan.yml	98
09_compressor.yml	99
10_generator.yml	100
11_UPS.yml	101
12_boiler.yml	102
13_pressure_vessel.yml	104
14_pressure_piping.yml	105
15_instrumentation.yml	106
Folder: B-role_class_properties	115
01_pump_role.yml	115
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 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       oneOf:
180         - type: null
181         - type: string
182           enum:
183             - "record is a duplicate"
184             - "removed or had departed"
185             - "lost"
186             - "record added by mistake"
187
188     date_of_actual_removal_or_departure:
189       description: is the date that the entity referent of the record was removed
190       ↪ or had departed
191       oneOf:
192         - type: null
193         - type: string
194           format: "date"
```

```
194
195     date_of_record_retirement:
196       description: is the date when the record was retired
197       oneOf:
198         - type: null
199         - type: string
200           format: "date"
201
202 content_entity_record_retirement_def:
203   properties:
204
205     record_is_active:
206       type: Boolean
207
208     retired_by:
209       oneOf:
210         - type: null
211         - $ref: "../08_person.yml"
212
213     reason_for_retirement:
214       read-only: true
215       oneOf:
216         - type: null
217         - type: string
218           enum:
219             - "record is a duplicate"
220             - "out-dated"
221             - "inaccurate"
222
223     date_of_record_retirement:
224       description: is the date when the record was retired
225       oneOf:
226         - type: null
227         - type: string
228           format: "date"
229
230 # DUPLICATION HANDLING
231
232 duplicate_record_def:
233   oneOf:
234     - type: null
235     - type: array
236       items:
237         type: object
238         $comment: the object may other asset, role, space, ... records, depending
239 ↪ on the context of where this property is used.
240
241 # METERS
242
243 meter_condition_definition:
244   properties:
245
246     meter:
```

```
246     description: is a selection of a pre-defined meter object.
247     $ref: "../41_meter.yml"
248
249     numeric_interval_value:
250       oneOf:
251         - type: null
252         - type: number
253
254     characteristic_trigger_value:
255       oneOf:
256         - type: null
257         - type: string
258
259 # COMPLIANCE REQUIREMENT DEFINITION
260
261 compliance_requirement:
262   properties:
263
264     name:
265       type: string
266       $comment: |
267         For example, 'ANSI Z358.1-2014 on weekly inspection of self-contained
↵ emergency wash equipment'.
268
269     requirement_detail:
270       description: provides all relevant descriptions
271       type: string
272       $comment: |
273         The following is an example of the requirement detail text for a
↵ compliance requirement.
274         applicable_asset_class:
275           - emergency eye-wash
276         requirement_source(s):
277           - ANSI Z358.1-2014 / Emergency Eyewash & Shower Standard / 4 Emergency
↵ Showers / 4.6 Maintenance and Training
278           ...
279           - ANSI Z358.1-2014 / Emergency Eyewash & Shower Standard / 4 Emergency
↵ Showers / 4.5 Installation
280           ...
281         source_content_guide:
282           - ANSI Z358.1-2014 4.6 states the requirement to check that shower
↵ still meets standards
283           - ANSI Z358.1-2014 4.5 states the standards to apply for ht check
284         perform_every:
285           - year
286
287     requirement_compliance_class:
288       description: indicates the level of compliance, with legislative being the
↵ top
289       $ref: "../compliance_class"
290
291 compliance_class:
292   type: string
```



```
293   enum:
294     - legislative
295     - corporate policy
```

01_asset.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: asset
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/01_as
6  type: object
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
14      ↪ asset record.
15
16    name:
17      type: string
18      description: The human readable short description of the asset.
19      $comment: |
20        Assumption: an non-is a specific commercial product is always built on site
21        ↪ for a specific purpose, and would permanently occupy a role. An example is an
22        ↪ aeration tank.
23
24    class:
25      $ref: "../B-entity_class_object_schema/01_asset_item_tool_class.yml"
26      description: indicates the class to which this asset is an instance.
27
28    GIS_object_ID:
29      type: string
30      integration: GIS
31      description: is unique ID of the GIS asset record, which represents the same
32      ↪ asset as this record.
33
34    specification_data:
35      type: object
36      description: is a set of specification property data. (The applicable properties
37      ↪ are dependent on the definition made in the chosen class.)
38
39    inferred_classes:
40      oneOf:
41        - type: null
42        - type: array
43          items:
44            $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 role,
↪   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 role, 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 role
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
↪   the "assigned_to_asset_role", "assigned_to_tool_user_group", or
↪   "assigned_to_tool_user_group" property. If the value here is "not assigned",
↪   then we know the asset is not assigned to anything. If the value here is null,
↪   we do not know whether this asset is assigned to anything.
114
115   assigned_to_asset_role:
116
117     oneOf:
118       - $ref: "./02_role.yml"
119       - type: null
120     description: Role that the asset is designated to play. This value persists even
↪   if the asset is temporarily removed from the location of the role (for reasons
↪   such as repair).
121
122   assigned_to_tool_user_group:
123     oneOf:
```

```
124     - $ref: "./04_org-chart_group.yml"
125     - type: null
126     description: A group of people, such as a facility, work area, or crew to whom
↪ the asset is assigned for use. Indicates the assignment of an asset (usually a
↪ tool) that does not have a system role.
127
128     assigned_to_tool_user:
129       oneOf:
130         - $ref: "./08_person.yml"
131         - type: null
132       description: Indicates the assignment of an asset (usually a tool) that does not
↪ have a system role.
133
134 # LOCATION
135
136     installation_or_parking_location:
137       oneOf:
138         - $ref: "./03_space.yml"
139         - type: null
140
141     service_address_or_coordinate:
142       oneOf:
143         - $ref: MaximoServiceAddressObject
144         - type: null
145       $comment: |
146         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 role
↪ 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 # 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_role".properties.name
292     # NOTE: actual script should contain additional condition handle formatting of
↪ 293     the name text when there is missing data in any concatenated property.
294     status:
295     checked: 2024-08-15
296
297 - name: Exclusion Of Part And Material Classes From Asset Classification
298   involves_prop: [class]
299   spec_ID: V15NNHZuxl
300   type: [validation, UI]
301   specification: |
302     Assertion Part:
303     For all assets "asset_x",
304     the value of (asset_x.class.properties.only used as a part asset) must be
↪ 305     FALSE
306     UI Part:
307     In all asset classification search or selection screens, eliminate or filter
↪ 308     out all classes "class_y",
309     where (class_y.properties.only used as a part asset) is TRUE
310     checked_on: 2024-08-15
311
312 - name: Valid Values of owned_by_the_group Property
313   involves_prop: [owned_by_the_group]
314   spec_ID: 410N2dr_xx
315   type: [validation, UI]
316   specification: |
317     - The valid range of values for selection includes the first or second of the
↪ 318     org-chart group hierarchy, specified in the
↪ 319     (/TWDM/3-System_Hierarchies/04_org_hierarchy.md) . For examples,
320     - first level example: York Region,
321     - second level example: Toronto Water
322     - The UI must only present the valid range of values to the users for
↪ 323     selection, and the valid range of values must be presented as a hierarchy.
324     checked_on: 2024-08-19
325
326 - name: Inheriting The Asset's Maintenance And Operator Group Values From Its Role
327   involves_prop: [operator_group, maintenance_group]
328   spec_ID: VJ1QRgIclg
329   specification: |
330     - if the value of asset_x.properties.assigned_to_asset_role is role_y, then
331     inherit the value of
332     - asset_x.properties.maintenance_group
333     - asset_x.properties.operator_group
334     from the same properties of role_y
335     checked_on: 2024-08-20
336
337 - name: Rendering of assignment_type Data Field.
338   involves_prop: [assignment_type]
339   spec_ID: 4yARRuvOex
340   type: [UI]
341   description: |
342     the options of this property should be presented as radial button
343     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 role", then
344       - asset_x.properties."assigned_to_asset_role" 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_role" 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_role" 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_role" 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
376   ↪ properties."is_a_commercially_available_product"
377   involves_prop: [is_a_commercially_available_product]
378   spec_ID: 4Jg2gYS0ee
379   type: [assertion]
380   specification: |
381     - Upon record creation, set the value to TRUE.
382     - Upon a asset_x.properties.class value change or a re-run of the Maximo rule
383     ↪ processor,
384       if asset_x.properties.class.properties."non-manufactured" = TRUE;
385       set the value to TRUE;
386       else set the value to FALSE.
387   status: [specified]
388
389 - name: Default Value of is_a_tool
390   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: NyFFWlUcll
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 role, 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_role"
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_role"
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 role, 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_role" = 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 Role
506 spec_ID: 4yBXuH8qle
507 type: [validation]
508 specification: |
509     if (asset_x.properties."assigned_to_asset_role") is NOT null
510     then
↪ (asset_x.properties."assigned_to_asset_role".properties.class.properties."discrete
↪ asset role") = TRUE
511     errorMessage: an asset can only be assigned to a discrete asset role
512
513 - name: inheriting the asset location information from its role
514 spec_ID: NJdGTHLqeg
515 type: [assertion]
516 specification: |
517     For an asset, asset_x, if
518     all of the following are true:
519         - asset_x.properties."operating_state" = "installed"
520         - asset_x.properties."assigned_to_asset_role" is NOT null
521     and one of the following is true
522         -
↪ asset_x.properties."assigned_to_asset_role".properties."asset_installation_location"
↪ is NOT null
523         -
↪ asset_x.properties."assigned_to_asset_role".properties."service_address_or_coordinate"
↪ is NOT null

```

```
524         then
525             (asset_x.properties."installation_or_parking_location") would be set to
↪ the value of
526
↪ (asset_x.properties."assigned_to_asset_role".properties."asset_installation_location")
527             (asset_x.properties."service_address_or_coordinate") would be set to the
↪ value of
528
↪ (asset_x.properties."assigned_to_asset_role".properties."service_address_or_coordinate")
529
530 - name: over-write of manufacturer and model information
531   spec_ID: 01J5RPPEKJCP11NBRW3A4XCKF7
532   specification: |
533       if the value of either
534   status: TBS
535
536 - name: a commercially available asset must be associated with manufacturer and
↪ model information
537   spec_ID: 4y3dRfLcee
538   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_role.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: role
4  $id:
↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/02_as
5  type: object
6
7  properties:
8
9      ID:
10         type: string
11         description: is the unique ID of the role known as the "tag number" or "entity
↪ number" in Avantis's vocabulary. (Avantis is the a WMS).
12
13     name:
14         type: string
15         description: is a structured description of the role.
16
17     parent:
18         $ref: "../02_role.yml"
```

```
19     description: references the role that is served by the larger asset, which
↪ physically subsumes the asset serving this role.
20
21 GIS_object_ID:
22     type: string
23     integration: GIS
24     description: is unique ID of the GIS record representing the same pumping
↪ station, chamber, or ... as this *role* record in Maximo.
25
26     class:
27         $ref: "B-entity_class_object_schema/02_role_classification.md"
28         description: denotes the broad types of asset that may play the role (e.g.,
↪ "breaker role"). It also denotes the useful function provided by an asset in the
↪ role to the larger system (e.g., the "tie-breaker role" provides tie-breaking
↪ function).
29
30     inferred_classes:
31         oneOf:
32             - type: null
33             - type: array
34               items:
35                 $ref: "B-entity_class_object_schema/02_role_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     role_status:
47         type: string
48         enum:
49             - specified
50             - active
51             - eliminated
52         description: is the life-cycle status of a role.
53         $comment: |
54             "specified" means the role is conceived and exists in some specification or
↪ design documentation; "active" means the necessary supports exist for an asset
↪ to serve in the role 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 space in which the asset serving the role would be
↪ installed.
```

```
61
62   serving:
63     $ref: "../02_role.yml"
64     description: identifies the asset this role is serving (by its role). For
↪ example, given a motor starter role, the value in this data field identifies the
↪ role of the motor controlled by that motor starter.
65
66   service_address_or_coordinate:
67     oneOf:
68       - $ref: MaximoServiceAddressObject
69       - type: null
70     description: is the geo-coordinate or the nearest street address of the asset.
71
72 # OWNER, OPERATOR, MAINTAINER GROUP
73
74   operator_group:
75     oneOf:
76       - type: null
77       - $ref: "../04_org-chart_group.yml"
78     description: group responsible for the operation of the asset in the role.
79
80   maintenance_group:
81     oneOf:
82       - type: null
83       - $ref: "../04_org-chart_group.yml"
84     description: group responsible for the overall maintenance of the asset - for
↪ example, a unit, work area, or crew.
85
86   #note:
87     # on: the absence of of a owned_by_the_group property for roles
88     # content: assets that are occupying a role are presumed to be owned by the
↪ City.
89
90   inherit_operator_group_value:
91     type: boolean
92     default_value: true
93
94   inherit_operator_group_from_parent:
95     type: boolean
96     default_value: true
97
98   inherit_maintenance_group_from_parent:
99     type: boolean
100     default_value: true
101
102   operational_criticality:
103     oneOf:
104       - $ref: "#/definitions/criticalityRatingDef"
105       - type: null
106     description: A role bears high operational criticality if the loss of the asset
↪ in the role will either reduce throughput or product quality (but not product
↪ safety) of the larger system.
107
```

```
108   protective_function_criticality:
109     oneOf:
110       - $ref: "#/definitions/criticalityRatingDef"
111       - type: null
112     description: A role 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.
113
114 # RECORD PROPERTIES
115
116 duplicate_record_of:
117   $ref: "./00_common_definitions.yml#/definitions/duplicate_record_def"
118
119 record_retirement_information:
120   $ref: "./00_common_definitions.yml#/definitions/record_retirement_def"
121
122 #####
123 # LOCAL OBJECT DEFINITION
124 #####
125
126
127 definitions:
128   criticality rating definition:
129     type: object
130     properties:
131       rating:
132         type: integer
133       description:
134         type: string
135     enum:
136       - rating: 1
137         description: TBD
138       - rating: 2
139         description: TBD
140       - rating: 3
141         description: TBD
142       - rating: 4
143         description: TBD
144       - rating: 5
145         description: TBD
146
147 #####
148 # RULES
149 #####
150
151
152 rule_spec:
153
154   - name: ID of Linear Assets Represented as Role in Maximo
155     spec_ID: Vku-67dDxx
156     involves_prop: [ID]
157     type: [triggered action]
```



```
158     specification:
159       trigger: replication creation of assetY record from the TWAG
160       action: apply Facility ID from TWAG as ID
161     status: false
162
163 - name: ID of Linear Assets Represented as Role in Maximo
164   spec_ID: Vku-67dDxx
165   involves_prop: [ID]
166   type: [triggered action]
167   specification:
168     - if:
169       oneOf:
170         - assetRoleClass:
171           properties:
172             className:
173               const: pumping station
174         - assetRoleClass:
175           properties:
176             className:
177               const: water treatment plant
178         - assetRoleClass:
179           properties:
180             className:
181               const: large chamber
182       then:
183         required: GIS_object_ID
184   status: false
185
186   #[]RULE VkiDyJcSxg: Before a role can be eliminated, all children, as well as
↵ the descendants of the role in the hierarchy must also be eliminated. $comment:
↵ a procedure should to be created to allow the recursive elimination of a role
↵ and all of its children.
187
188   #[]RULE VygDCOFrx1: When a role is "eliminated", it must no longer be visible in
↵ any view of the role hierarchy. (Its entire branch must not be available either,
↵ because all of its descendants would be eliminated as well.)
189   #[]Review with To ASMP: with this rule, we no longer need to have a hierarchy
↵ branched for retired roles.
190
191   #[]RULE EkP5qy5Sx1: If change auditing cannot be turned on, then when a role
↵ record status is "eliminated", all the specification in the record's data fields
↵ must be frozen.
192
193   #[]RULE:
194   # IN COMMON LANGUAGE: At any given time, each role may only have a single asset
↵ assigned to it (i.e., associated with the role via the asset's
↵ "assigned_to_asset_role" property).
```

03_space.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: space
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 space that completely contains this space.
20
21   name:
22
23     type: string
24     description: Short name. Should be one that is commonly use by staff in
25     ↪ communication. For example, "boardroom"
26
27   complete_name:
28
29     type: string
30     read-only: true
31     rule_spec:
32       - spec_ID:
33         name: complete name generation
34         id: 01JFVNSOYDFD7K5DP4NVMSKTY8
35         status:
36         $comment: |
37           Automatically generated by the system and not editable. The value is name to
38           ↪ that of its parent, its grandparent ... all the way up that facility space.
39
40   enclosed_by_asset:
41
42     oneOf: [$ref: "./01_asset.yml", type: null]
43     description: indicates that the space is what is enclosed by (and immediately
44     ↪ surrounding) the asset, such as a
45       - building,
46       - structural tank,
47       - equipment cabinet,
48       - 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
↪   are dependent on the definition made in the chosen class.)
55
56 inferred_classes:
57
58   oneOf:
59     - type: array
60       items:
61         $ref: "../B-entity_class_object_schema/03_space_class.yml"
62     - type: null
63   read-only: TRUE
64
65 service_address_or_coordinate:
66
67   oneOf:
68     - $ref: MaximoServiceAddressObject
69     - type: null
70   $comment: this is referencing Maximo's native service address object
71
72 status:
73
74   type: string
75   enum:
76     - specified
77     - realized
78     - eliminated
79   $comment: |
80     This field allows the user to specify whether the space is merely specified,
↪   or whether the boundary enclosing the space have been constructed (or
↪   alternatively, the fiat property / or area boundary around is formally
↪   established and approved.) - i.e. "exists".
81     Note that an "eliminated" space should be removed from the hierarchy, and
↪   should not be visible for users conducting maintenance, reliability, planning
↪   and scheduling functions.
82     The term "eliminated" is chosen to indicate that space disappear by the fact
↪   that object bound or defined the space, such as wall, ceilings, or property
↪   lines are removed.
83
84 confined_space:
85
86   type: boolean
87
88 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     # 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
7  properties:
```

```
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 # RECORD PROPERTIES
```

```
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:
37                 the value of item_x.properties.name would be the semi-colon delimited
38             ↪ 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 role 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" "role", "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
222             description: The expected of the item in the order format
223             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     # 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/content_entity_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: V1ulHHWQgx
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 # 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:
78 ↪   "../00_common_definitions.yml#/definitions/content_entity_record_retirement_def"
79
80 $comment: ASMP does not expect the tool item master to be widely used during the
81 ↪   initial adoption of Maximo - we expect that most tools would initially be
82 ↪   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   # 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:
17 ↪    "../00_common_definitions.yml#/definitions/content_entity_record_retirement_def"
18
19 # 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
↵ works for
32     oneOf:
33         - type: null
34         - type: string
35
36 status:
37     type: string
38     enum:
39         - active
40         - inactive
41         - departed
42
43 person_roles:
44
45     description: The role, associated with a certain type of system access, that the
↵ person plays in the system.
46     oneOf:
47         - type: null
48         - type: array
49         items:
50             $ref: MaximoSecurityRole
51
52 # NOTE: persons' association to trade in documented in the trade object
53
54 qualifications:
55
56     description: The qualifications that the person gained through training or
↵ certification.
57     oneOf:
58         - type: null
59         - type: array
```

```
60     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     # 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  # RECORD PROPERTIES
36
37    duplicate_record_of:
38      $ref: "../00_common_definitions.yml#/definitions/duplicate_record_def"
39
40    record_retirement_information:
41      $ref:
42        ↪ "../00_common_definitions.yml#/definitions/content_entity_record_retirement_def"
```

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

```
50
51     trade:
52       $ref: "../10_trade.yml"
53
54     skill_level:
55       type: string
56       $comment: load the list of skilled trades
57
58   status:
59     type: string
60     enum:
61       - active
62       - inactive
63
64 # 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_wa
5 type: object
6
7 properties:
8
9   ID:
10     description: the unique identifier for the warranty contract
11     type: string
12
13   description:
14     description: a brief summary of the warranty contract
15     type: string
16
17   vendor:
18     description: identifies the vendor or provider of the warranty
19     $ref: MaximoCompanyObject
20
21   warranty_start_date:
22     description: the first effective day of the warranty
23     oneOf:
24       - type: null
25       - type: string
26         format: date
27
28   warranty_expiration_date:
29     description: the last effective date of the warranty
30     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_roles:
43     description: the list of roles or more specifically the assets installed in the
↪   roles covered by the warranty
44     oneOf:
45       - type: null
46       - type: array
47         items:
48           $ref: "../02_role.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 # 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/content_entity_record_retirement_def"
```

12_service_contract.yml

```
1 $schema: http://json-schema.org/draft-07/schema#
2 title: Service Contract
3 $id:
4   ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/1-Schemas/A-entity_record_schema/13_se
5 type: object
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_roles:
35    description: the list of roles or more specifically the assets installed in the
36    ↪ roles covered by the contract
37    oneOf:
38      - type: null
39      - type: array
40        items:
41          $ref: "./02_role.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:
```

```
48     $ref: "./01_asset.yml"
49
50 covers_work:
51   description: identifies work (job plans) the contractor would perform.
52   oneOf:
53     - type: null
54     - 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 # 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/content_entity_record_retirement_def"
```

32_job_plan.yml

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

```

25 # JOB PLAN APPLICABILITY NOTES
26
27 specific_to_asset_classes:
28   description: identifies the asset classes on which the activity specified in
↪   this job plan can be done.
29   type:
30     oneOf:
31     - 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_role_classes:
37   description: identifies the role classes - more specifically, the assets
↪   installed in these roles - 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_role_classification.md"
44   $comment: >
45     Examples of a role class include the tie-breaker and effluent turbidity meter
↪   role 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_roles:
69   description: identifies the roles 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_role.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 role, 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 # 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/content_entity_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 operational_unit:
18   description: indicates Toronto Water's operational unit, on the org-chart group
19   ↪ hierarchy, such as Waste and Storm Water Pumping (symbol - WASP) or Humber
20   ↪ Wastewater Treatment Plant (symbol - THR).
21   $ref: "./04_org-chart_group.yml"
22 supervisor_group:
23   description: indicates a sub-group of the operational unit, that is led by a
24   ↪ supervisor who is accountable for the performance of the specified work.
25   $ref: "./04_org-chart_group.yml"
26 crew_assignment:
27   description: identifies a crew, under the supervisor group, that is always
28   ↪ assigned to perform the specified work
29   oneOf:
30     - type: null
31     - $ref: "./04_org-chart_group.yml"
32   comment: This value will be determined by (and must be consistent with) the
33   ↪ maintainer_org-chart_group value - situated at a lower level of the org-chart
34   ↪ group hierarchy
35 member_of_PM_set:
36   description: indicates that this PM is a member of a set of related PMs. For
37   ↪ example, the PMs for raw water pump 1, 2, 3 are all members of a PM set named
38   ↪ Raw Water 5-year Disassembly Maintenance.
39   oneOf:
40     - type: null
41     - $ref: "#/definitions/PM_set"
42   rule to add []: only applicable to higher-level PM
43
44 Avantis_PM:
45   description: indicate the Avantis PM (the legacy WMS) that this Maximo PM
46   ↪ (equivalent to an Avantis PM task) was a part of.
47   oneOf:
48     - type: null
49     - type: string
50   read-only: TRUE
51   comment: This field can be eliminated in the future.
52   work_entity_harmonization: WR(x), WO(x), JP(x)
53
54 processes_covered_by_PM:
55   description: a list of all major process systems covered by the work specified
56   ↪ in this PM.
57   read-only: TRUE
58   oneOf:
59     - type: null
60     - type: object
61       properties:
62         ranking:
63           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 role_to_work_on:
67   description: indicates role at which the specified must be performed.
68   oneOf:
69     - type: null
70     - $ref: "../A-entity_record_schema/02_role.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/role.
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       role:
103         oneOf:
```

```
104         - type: null
105         - $ref: "./02_role.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
116     ↪ the PM. Instead they could be recorded as contributory work in the job plan
117     ↪ route
118
119     estimated_duration:
120         description: is the estimated time to complete the activity in the job plan
121         $ref: "./00_common_definitions.yml#/definitions/frequency_interval_definition"
122
123     parts_or_material_requirements:
124         description: identifies the parts or material required to complete a work order
125         ↪ generated from the PM.
126         oneOf:
127             - type: null
128             - type: array
129             items:
130                 $ref:
131                 ↪ "./00_common_definitions.yml#/definitions/item_requirement_definition"
132
133     tool_requirements:
134         description: identifies the tools required to complete a work order generated
135         ↪ from the PM.
136         oneOf:
137             - type: null
138             - type: array
139             items:
140                 $ref:
141                 ↪ "./00_common_definitions.yml#/definitions/tool_requirements_definition"
142
143     skill_and_trade_requirements:
144         description: identifies the trades and qualifications of each trade needed to
145         ↪ complete the work.
146         oneOf:
147             - type: null
148             - type: array
149             items:
150                 $ref: "./00_common_definitions.yml#/definitions
151                 ↪ trade_requirement_definition"
152
153     service_requirements:
154         description: identifies (contracted) service needed to complete to complete a
155         ↪ work order generated from the PM.
156         oneOf:
```

```
148     - type: null
149     - type: array
150       items:
151         $ref:
152 ↪     "./00_common_definitions.yml#/definitions/service_requirement_definition"
153
154     # Work Triggering Specification
155
156     # Note: the specification is not complete for data mapping purposes, it is
157 ↪     complete for requirement gather
158
159     next_due_date_based_on:
160       type: string
161       enum:
162         - work start date
163         - work completion date
164
165     trigger_condition: # aka work generation condition
166       oneOf:
167         - type: null
168         - $ref: "#/definitions/time-based_trigger_specification"
169         - $ref: "#/definitions/meter-based_trigger_specification"
170
171     # LEGISLATIVE DESIGNATION
172
173     compliance_level:
174       $ref: "./00_common_definitions.yml#/definitions/compliance_class"
175     work_entity_harmonization: WR(x), WO(_), JP(_)
176
177     # RECORD PROPERTIES
178
179     duplicate_record_of:
180       $ref: "./00_common_definitions.yml#/definitions/duplicate_record_def"
181
182     record_retirement_information:
183       $ref:
184 ↪     "./00_common_definitions.yml#/definitions/content_entity_record_retirement_def"
185
186     #####
187     # LOCAL OBJECT DEFINITIONS
188     #####
189
190     definitions:
191
192       time-based_trigger_specification:
193         next_due_date:
194           type: string
195           format: date
196         frequency_interval:
197           $ref: "./00_common_definitions.yml#/definitions/frequency_interval_definition"
```

```
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     role_to_work_on:
76
77     description: indicates role at which the specified work is to be performed.
78     oneOf:
79       - type: null
80       - $ref: "../A-entity_record_schema/02_role.yml"
81     #[] should have name and description, like asset_to_work_on
82
83     rule_spec:
84       - name: mutual exclusion of asset_to_work_on and role_to_work_on values
85       spec_ID: 01JFK49T43T1TF3HBTBTDPMN5Y
```

```
86     form: short
87     specification: |
88       one, but only one, of the following properties can have a non-null value:
89       - role_to_work_on
90       - 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 be represented.
110
111   asset_or_asset_role_criticality:
112
113     description: is the criticality values inherited from the asset or role being
114     ↪ 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/role.
```

```
137   oneOf:
138     - type: null
139     - type: object
140       properties:
141
142         sequence:
143           type: number
144
145         asset:
146           oneOf:
147             - type: null
148             - $ref: "./01_asset.yml"
149
150         role:
151           oneOf:
152             - type: null
153             - $ref: "./02_role.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 # FAILURE REPORTING
161
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
```

```
188     specification: []
189
190     $comment: multiple values are allowed
191
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: 01JFH2NE68WONCSBKKSYPZRA2Z
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 role
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
339 crew_assignment:
340
341   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 role) 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 role) 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
432     ↪ from the PM.
433     oneOf:
434       - type: null
435       - type: array
436         items:
437           $ref:
438     ↪ "./00_common_definitions.yml#/definitions/tool_requirements_definition"
439
440   service_requirements:
441
442     description: identifies (contracted) service needed to complete to complete a
443     ↪ work order generated from the PM.
444     oneOf:
445       - type: null
446       - type: array
447         items:
448           $ref:
449     ↪ "./00_common_definitions.yml#/definitions/service_requirement_definition"
450
451   skill_and_trade_requirements:
452
453     description: identifies the trades and qualifications of each trade needed to
454     ↪ complete the work.
455     oneOf:
456       - type: null
457       - type: array
458         items:
459           $ref: "./00_common_definitions.yml#/definitions
460     ↪ trade_requirement_definition"
461
462 # DATES
463
464   issue_reported_date:
465
466     description: is the date that the issue or failure was reported.
467     oneOf:
468       - type: null
469       - type: string
470         format: date
471
472   work_requested_date:
473
474     description: is the date when the work request was submitted.
475     oneOf:
476       - type: null
477       - type: string
478         format: date
```

```
475 request_approval_date:
476
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
535     ↪ either a cancelled work order or work request).
536     oneOf:
537       - type: null
538       - type: string
539         format: date
540
541   WO_closing_date:
542
543     description: is the date when the work order was closed.
544     oneOf:
545       - type: null
546       - type: string
547         format: date
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
555     ↪ example, the PMs for raw water pump 1, 2, 3 are all members of a PM set named
556     ↪ Raw Water 5-year Disassembly Maintenance.
557
558     oneOf:
559       - type: null
560       - $ref: "#/definitions/PM_set"
561
562   parent_work_order:
563
564     description: indicates the more comprehensive PM, usually a shut-down PM, that
565     ↪ this PM is a part of.
566     oneOf:
567       - type: null
568       - $ref: "../A-entity_record_schema/33_PM.yml"
569
570   WO_only: true
571
572   rule_spec:
573     name: Work Type of Descendant Work Orders
574     form: short
575     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   # 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:
665 ↪ "../00_common_definitions.yml#/definitions/content_entity_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 # 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   role:
27     oneOf:
28       - $ref: "../02_role.yml"
29       - type: null
30     description: identifier of the role 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 # 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   synonym_names:
16     oneOf:
17       - type: null
18       - type: array
19         items:
20           type: string
21     description: other synonymous names that may be used by a user in search
22
23   class_definition:
24     oneOf:
25       - type: null
26       - type: string
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,
  ↪ role, space, etc).
32
33   # 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/content_entity_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_object_schema.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: NJ1E1Zb0gg
32         status: TBS
33         $comment: A false or null value materially mean the same thing.
34
```

02_role_class.yml

```
1 ---
2 $schema: "http://json-schema.org/draft-07/schema#"
3 title: role classification object
4 $spec_ID: B-entity_class_object_schema/02_role_classification.md
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"
```

```
15     #inherit the definitions and rules from th
16     ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
17
18 parent_class:
19     $ref: B-entity_class_object_schema/02_role_classification.md
20     sort order: 1-30
21
22 discrete_asset_role:
23     oneOf: [type: boolean, type: null]
24     read-only: true
25     description: a role that can be occupied by a single discrete asset.
26
27 functional_structure_role:
28     oneOf: [type: boolean, type: null]
29     read-only: true
30     description: a role that cannot be occupied any discrete asset, but can be the
31     ↪ parent to other roles.
32
33 defined_set_of_roles:
34     oneOf: [type: boolean, type: null]
35     read-only: true
36     description: a role that can be occupied by a collection of discrete assets.
37
38 compatible_asset_occupant_classes:
39     oneOf:
40         - type: array
41           items:
42             $ref: "./01_asset.yml"
43         - type: null
44     $comment: |
45         [ ]RULE NJQ6BwsVee: A asset must be an instance of one of classes listed in
46     ↪ this field to be allowed to occupy a role under this class.
47
48 #####
49 # 2. High level rules
50 #####
51
52 rule_spec:
53     - name: Is an role, functional structure role, or defined set of roles
54       spec_ID: NyD4XGbuex
55       specification: |
56         if roleClassX is a descendent of "Discrete Asset Role" in the role
57     ↪ classification hierarchy:
58         set roleClassX.properties."a discrete asset role" to TRUE
59         set roleClassX.properties."a functional structure role" to FALSE
60         set roleClassX.properties."a defined ser of role" to FALSE
61         elif roleClassX is a descendent of "functional structure role" in the role
62     ↪ classification hierarchy:
63         set roleClassX.properties."a discrete asset role" to FALSE
64         set roleClassX.properties."a functional structure role" to TRUE
65         set roleClassX.properties."a defined ser of role" to FALSE
66         elif roleClassX is a descendent of "Defined Set of Roles" in the role
67     ↪ classification hierarchy:
```

```
62     set roleClassX.properties."a discrete asset role" to FALSE
63     set roleClassX.properties."a functional structure role" to FALSE
64     set roleClassX.properties."a defined ser of role" to TRUE
65     status: specified
```

03_space_class.yml

```
1  title: role classification object
2  $id: spaceClassObject
3  $schema: "http://json-schema.org/draft-07/schema#"
4  type: object
5
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: spaceClassObject
```

04_org_class.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: role classification object
4  $id: orgGroupClassObject
5  type: object
6
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: orgGroupClassObject
```

32_discrete_activity_class.yml

```
1 ---
2 title: discrete activity classification object
3 $id: discreteActivityClassObject
4 $schema: "http://json-schema.org/draft-07/schema#"
5 type: object
6
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: discreteActivityClassObject
```

33_work_type.yml

```
1 title: work type object
2 $id: workOrderClassObject
3 $schema: 'http://json-schema.org/draft-07/schema'
4 type: object
5
6 #placeholder
```

Folder: 2-Classification_Trees

01_asset_classification.md

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

02_role_classification.md

```
1 ## Top-Level Role Classification
2
3 * Discrete Asset Role*
```

```
4      * generator role
5          * backup generator role
6          * emergency generator role
7      * breaker role
8          * bus feeder breaker role
9          * line protection breaker role
10         * load breaker role
11         * tie-breaker role
12 * Collection of Assets Role*
13     * functional structure role
14         * system block
15         * facility
16             * pumping station
17             * water treatment facility
18             * wastewater treatment facility
19             * lab
20             * yard
21     * process
22 * Linear Functional Structure Role*
23     * system train role
24     * line role
25         * simple line role
26         * primary path line role
27     * junction role
28 * Defined Set of Roles*
29     * defined set of discrete assets
30     * defined set of functional structures
31
32
33 ## Requirements for Implementer
34
35 * []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"
36
37 ## Notes
38
39 * []TODO #TW: the following should be moved to the asset classification.
40     * system on a skid
41     * system of standardized modular parts
```

03_space_classification.md

```
1 ## Top-Level Space Classification and Examples
2
3 * Discrete Space*
4     * building or structure interior
5     * room interior
6     * corridor
7     * mezzanine
8     * stairwell
```

```
9      * stairwell segment
10     * tunnel interior
11     * equipment cabinet interior
12     * storage cabinet interior
13     * facility site space
14     * facility site section
15     * building or structure exterior
16     * vehicle interior
17 * Collection of Spaces*
18     * spaces of a building or structure [^1]
19     * define set of spaces
20
21 ## Requirements for Implementer
22
23 []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"
24
25 ## Notes
26 ### Footnotes
27 [^1]: a collection of indoor and outdoor spaces inside and around a structure.
  ↳ []TODO #TW "add to the TWONTO".
```

04_org_classification.md

```
1 ## Top-Level Org/Group Classification
2
3 * Level of Government*
4     * provincial government
5     * regional government
6     * municipal government
7 * Group in the City*
8     * cluster
9     * division
10    * section
11    * unit
12    * Group in TW*
13        * work area
14        * crew
15 * government agency
16 * private business
17 * non-governmental organization
18
19 ## Requirements for Implementer
20
21 []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  * corrective
4  * emergency [^1]
5  * investigative
6  * preventive
7  * informational
8  * project
9    * contractor support
10
11 ## Notes
12 [^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).
13 [^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  * Condition Evaluation*
3    * quick check
4    * inspection and evaluation
5    * test and analysis
6    * condition analysis
7  * sample collection
8  * cause investigation
9  * repair or service [^1]
10    * calibration
11    * asset replacement
12    * asset part replacement
13    * asset part movement
14  * Move or Replace*
15    * new asset installation
16    * asset movement
17    * asset part movement
18    * asset replacement
19    * asset part replacement
20    * asset hand-over
21  * Life Cycle Events*
22    * asset commissioning
23    * asset hand-over [^3]
24    * final asset decommissioning
25  * Asset Modification*
26    * modify asset set-point
27    * physical modification to asset
28    * physical modification to building or structure [^2]
29  * asset assignment
```

```
30 * Contributory Work*
31 * item procurement
32 * work coordination
33 * safety preparation
34 * setup
35 * takedown of setup
36 * travel
37 * design or redesign
38 * creation of new role
39 * removal of existing role
40 * Asset Data*
41 * record information correction
42 * record retirement
43
44
45
46 ## Requirements for Implementer
47
48 ```yaml
49
50 rule_spec:
51 - name: Valid Assignment of an Asset
52   spec_ID: 01JDCNEFAED17CWF2K851ZAJKW
53   type: [assertion]
54   description: |
55     classes names specified with an asterisk symbol shall have the their
56     ↪ .property.can_be_applied_to_instances value set to false
57 ```
58
59
60 ### Footnotes
61 [^1]: more will be added before the final implementation.
62 [^2]: the physical modification of a building or a structure may result in the
63     ↪ creation and removal of a space, hence it is singled out.
64 [^3]: the process by which a asset's ownership is transferred from a capital project
65     ↪ to Toronto Water.
```

Folder: 3-System_Hierarchies

02_role_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: Drinking Water Treatment Plants, the
19    ↪ entity represents a set of things. In the case of the example - the set of
20    ↪ drinkin water treatment plants in TW.
21
22  ## Notes
23
24  [^1]: i.e. {[FCL]}, {[FIS]}, {[FHO]}, {[FHA]}
25  [^2]: the set of 18 pumping stations
26  [^3]: the set of all reservoirs and elevated tanks
27  [^4]: pumping station for waste and storm water
28  [^5]: the set of all wet-whether storage, inline an offline.
29  [^6]: i.e. {[THC]}, {[THR]}, {[TAB]}, {[TNT]}
```

03_space_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    ↪ spaces within the drinking water treatment plants in TW.
21
22  ## Notes
```

04_org_hierarchy.md

```
1 # Organizations in TW and Interact with TW
2
3 ## The Hierarchy
4
5 * Region of Durham
6 * York Region
7 * Peel Region
8 * Province of Ontario
9 * Metrolinx
10 * TRCA
11 * Envave Energy
12 * City of Toronto
13     * Solid Waste
14     * Toronto Water
15         * Distribution & Collection Section
16             * DOS
17             * WASP
18             * Central Services
19             * Program Maintenance
20         * Water Treatment & Supply Section
21         * Wastewater Treatment Section
22             * Ashbridge's Bay Wastewater Treatment Plant
23             * Highland Creek Wastewater Treatment Plant
24                 * Work Area 1
25                 * Work Area 2
26
27 ## Notes
28 * 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 space
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
52     ↪ L/s
53
54   pump_head:
55     oneOf:
56       - type: null
57       - type: number
58     description: Indicates the pressure head for the pump in metres
59
60   submersible:
61     oneOf:
62       - type: null
63       - 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  type: object
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
    ↪ specialized valve
34    enum:
35      - "Air release valve"
36      - "Backflow Preventer"
37      - "Check"
38      - "Pressure Relief Valve"
39      - "Solenoid"
40    $comment: may be redundant to label them valve in the valve type
41
42  size:
43    oneOf:
44      - type: null
45      - type: number
46    description: Indicates the valve size in inches
47
48  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
26    ↪ breaker in Volts
27
28    max_amperage:
29      oneOf:
30        - type: null
31        - type: number
32      description: Indicates what the maximum continuous current rating for the
33    ↪ breaker in Amps
34
35    main_contactor_type:
36      oneOf:
37        - type: null
38        - type: string
39      description: Indicates the ANSI type for the valve
40      enum:
41        - "air insulated"
42        - "air insulated, air blast"
43        - "vacuum insulated"
44        - "oil insulated"
45        - "gas insulated"

```

05_starter.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: starter
4  $id:
5  ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/05_starter.yml
6  type: object
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
7  type: object
8
9  $comment: >
10
11  properties:
12
13    primary_voltage:
14      oneOf:
15        - type: null
16        - type: number
17      description: Indicates the primary voltage rating for the transformer in Volts
18
19    secondary_voltage:
20      oneOf:
21        - type: null
22        - type: number
23      description: Indicates the secondary voltage rating for the transformer in Volts
24
25    power_rating:
26      oneOf:
27        - type: null
28        - type: number
29      description: Indicates the power rating for the transformer in kVA
30
31    oil_filled:
32      oneOf:
33        - type: null
34        - type: boolean
35      description: Indicates the requirement for oil cooling for the transformer
36
37    pressure_relay:
38      oneOf:
39        - type: null
40        - type: boolean
41      description: Indicates the presence of a sudden pressure relay
42
43    cooling_air_fan:
44      oneOf:
45        - type: null
46        - type: boolean
47      description: Indicates the presence of a cooling air fan(s)
48
49    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_HV
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    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
```

```
34
35 has_dehumidification_function:
36   oneOf:
37     - type: null
38     - type: boolean
39   description: Indicates the presence of a dehumidifier
40
41 has_maintainable_damper:
42   oneOf:
43     - type: null
44     - type: boolean
45   description: Indicates the presence of a maintainable damper
46   $comment:
47
48 uses_belt:
49   oneOf:
50     - type: null
51     - type: boolean
52   description: Indicates the presence of replaceable belts
53
54
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:
```

```
24     oneOf:
25       - type: null
26       - type: string
27     description: Indicates the type of drive coupling
28     enum:
29       - "direct drive"
30       - "belt drive"
31       - "gear drive"
32       - "flexible"
33       - "chain drive"
34       - "hydraulic"
35
36     bearings_are_sealed:
37       oneOf:
38         - type: null
39         - type: boolean
40     description: Indicates the presence of sealed or shielded bearings
41
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_compressor.yml
6  type: object
7  $comment: >
8
9  properties:
10
11    rated_pressure:
12      oneOf:
13        - type: null
14        - type: number
15      description: Indicates the capacity rating for the compressor in kPa
16
17    rated_flow:
18      oneOf:
19        - type: null
20        - type: number
21      description: Indicates the capacity rating for the compressor in SCM/H
22
23    drive_coupling:
24      oneOf:
25        - type: null
26        - type: string
27      description: Indicates the type of drive coupling
28      enum:
```

```
29     - "direct drive"
30     - "belt drive"
31     - "gear drive"
32     - "flexible"
33     - "chain drive"
34     - "hydraulic"
35
36   bearings_are_sealed:
37     oneOf:
38       - type: null
39       - type: boolean
40     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_generator.yml
6  type: object
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
14      ↪ suppression or any life safety systems
15
16    is_mobile:
17      oneOf:
18        - type: null
19        - type: boolean
20      description: Indicates if the generator is mobile
21
22    power_rating:
23      oneOf:
24        - type: null
25        - type: number
26      description: Indicates the power rating in kilo watts
27
28    voltage_rating:
29      oneOf:
30        - type: null
31        - type: number
32      description: Indicates the capacity rating for the generator in Volts
33
34    drive_type:
35      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
54     ↪ load
55
56   drive_coupling_type:
57     oneOf:
58       - type: null
59       - type: string
60     description: Indicates the type of drive coupling
61     enum:
62       - "direct drive"
63       - "belt drive"
64       - "gear drive"
65       - "flexible"
66       - "chain drive"
67       - "hydraulic"
68
69   bearings_are_sealed:
70     oneOf:
71       - type: null
72       - type: boolean
73     description: Indicates the presence of sealed of 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_UP
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
↪      suppression system, life safety system, or elevator of a high building systems
16
17    battery_capacity:
18      oneOf:
19        - type: null
20        - type: number
21      description: Indicates the power rating in kilo watt hours
22
23    voltage_rating:
24      oneOf:
25        - type: null
26        - type: number
27      description: Indicates the output voltage of the UPS in Volts
28
29    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:
↪   https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/12_bo
5  type: object
6
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:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/main/4-Class_Dependent_Specifications/13_pressure_vessel.yml
6  type: object
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
```



```
40     - type: string
41     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
22         ↪ in kPa
23
24     max_temperature:
25         oneOf:
26             - type: null
27             - type: number
28         description: Indicates the maximum medium temperature within the pressure piping
29         ↪ in degrees celsius
30
31     contained_medium:
32         oneOf:
33             - type: null
34             - type: string
35         description: Indicates the medium within the pressure piping
36         enum:
37             - "Water"
38             - "Steam"
39             - "Glycol"
40             - "Refrigerant"
41             - "Compressed Air"
42             - "Digester Gas"
43             - "Ozone"
44
45     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
8  properties:
9
10     parameter:
11       oneOf:
12         - type: null
13         - type: string
14       description: Indicates the parameter that is being measured
15       enum:
16         - "density"
17         - "flow rate"
18         - "humidity"
19         - "level"
20         - "current"
21         - "power"
22         - "position"
23         - "pressure"
24         - "speed"
25         - "temperature"
26         - "torque"
27         - "uv"
28         - "vibration"
29         - "weight"
30         - "specific gravity"
31         - "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           - "Milliamperere (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 ( $\mu\text{g}/\text{m}^3$ )"
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 role
4  $id:
5    ↪ https://raw.githubusercontent.com/TW-ASMP/TWDM/4-Class_Dependent_Specifications/B-role_class_properties/01_pump_role.yml
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 role 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 role
20
21    required_total_dynamic_head:
22      oneOf:
```

```
23     - type: null
24     - type: number
25     description: Indicates the pressure head for the pump in metres
26
27 must_be_submersible:
28     oneOf:
29         - type: null
30         - type: boolean
31     description: Indicates if the pump is submersible
```

Folder: 5-Functions