Toronto Water WMS Configuration Schema

Table of contents

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:
    ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/A-entity_record_schema/00_co
5
6
7   definitions:
8
9     #================================
10    # INVENTORY
11    #================================
12
13    stocked_at_TW_def:
14
15      type: boolean
16      description: a true value indicates that the item is a stocked item at the
    ↪  Toronto Water
17
18      TW_rule:
19        - name: default value of .properties.stocked_at_TW
20          spec_ID: EkxMAfT5ee
21          specification: |
22            For a given item_or_tool_x,
23              the value of item_or_tool_x.properties.stocked_at_TW is set to false by
    ↪  default
24
25   rotating_property_def:
26
27      type: boolean
28
29      $comment: |
30        For posterity: a rotating item is a trackable item - each instance of the item
    ↪  being a rotating asset record. When the value of this field is true, we must
    ↪  serialize every instance of the item. This commitment is beyond what we can
    ↪  presently achieve at TW. The more pragmatic starting point considered in 2024 is
    ↪  as follows:
31          1) for the initial implementation, specify all items as non-rotating
32          2) an non-rotating item definition, containing the mfr, model, and
    ↪  ordering_information, can be associated with any asset, through the the asset's
    ↪  item_product_master_record property.
33          2) in the future, if we wish to convert the non-rotating item definition to
    ↪  a rotating item definition, we would serialize all the assets associated to the
    ↪  item definition, and convert them to rotating assets.
34
35      rule_spec:
```

```yaml
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
   ↪  time
42            $comment:
43            status: specified
44
45    manufacturer_and_model_def:
46
47      type: object
48      properties:
49
50        manufacturer:
51          $ref: MaximoCompanyObject
52
53        product_model:
54          $ref: "#/definitions/product_model_def"
55
56    product_model_def:
57
58      type: object
59      properties:
60
61        model_and_sub-model:
62          oneOf:
63          - type: null
64          - type: string
65          description: For example, "Multilin 869"
66
67        version_or_model_year:
68          oneOf:
69          - type: null
70          - type: string
71          description: Identifies the specific version of the product model. For
   ↪  example "v2" or "2023".
72
73        manufacturer_PN:
74          oneOf:
75          - type: null
76          - type: string
77          description: The manufacturer designator identifying the exact product item.
78
79    plain-text_manufacturer_and_model_def:
80      type: object
81
82      $comment: |
83        The following is an example of the plain text manufacturer and model
   ↪  definition
84          manufacturer: General Electric
85          model_and_sub-model: Multilin 750
```

```yaml
86          version_or_model_year: 2024
87
88    #=================================
89    # FAILURE
90    #=================================
91
92    failure_code:
93
94      type: object
95      properties:
96
97        code:
98          type: string
99
100       name:
101          type: string
102
103       description:
104          type: string
105
106       failure_code_type:
107          type: string
108          enum:
109            - problem
110            - cause
111            - remedy
112
113       site:
114          type: object
115          $ref: MaximoSiteObject
116
117       failure_classes:
118          oneOf:
119            - type: null
120            - type: array
121              items:
122                type: object
123                $ref: MaximoFailureClass
124
125       status:
126          type: string
127          enum:
128            - draft
129            - approved
130
131    #=================================
132    # JOB PLAN RESOURCE
133    #=================================
134
135    item_requirement_definition:
136    #-------------------------
137      properties:
138
```

```yaml
139        item_reference:
140          $ref: "./05_item_master.yml"
141
142        required_quantity:
143          type: number
144
145        unit:
146          $ref: "#/definitions/unit_of_measure"
147
148   tool_requirements_definition:
149   #---------------------------
150     properties:
151
152        tool_reference: # reference for both stocked and un-stocked tool
153          $ref: "./06_tool_master.yml"
154
155        required_quantity:
156          type: number
157
158   service_requirement_definition:
159   #----------------------------
160     properties:
161
162        service_reference:
163          $ref: "./07_service_item_master.yml"
164
165        required_quantity:
166          type: number
167
168        unit:
169          type: string
170          enum: >
171            - hour
172            - instance
173
174   trade_requirement_definition:
175   #----------------------------
176     properties:
177
178        trade_type:
179          $ref: "../B-entity_class_object_schema/08_trade_type.yml"
180
181        required_quantity:
182          type: number
183
184        qualification_requirement:
185          oneOf:
186          - type: null
187          - type: array
188            items:
189              $ref: "../B-entity_class_object_schema/09_qualification.yml"
190
191   #==============================
```

```yaml
192    # UNIT OF MEASURE
193    #===============================
194
195    unit_of_measure:
196
197      description: Represents a unit of measure (UOM) used in inventory management to
       ↪  track quantities of items.
198
199      properties:
200
201        code:
202
203          description: is the unique identifier or code for the unit of measure.
204          type: string
205          $comment: e.g., "EA", "kg"
206
207        name:
208
209          description: is the full name of the unit of measure.
210          type: string
211          $comment: e.g., "Each", "kilogram"
212
213    #========================
214    # Record Retirement
215    #========================
216
217    record retirement definition:
218
219      properties:
220
221        record retired:
222          type: Boolean
223
224        reason for retirement:
225          oneOf: [type: string, type: null]
226          updated by system: true
227          sort order: 20-20
228          $comment: the value should be written by the system, from a asset or record
       ↪  retirement transaction.
229
230    frequency_interval_definition:
231
232      properties:
233
234        frequency_quantity:
235
236          type: number
237
238
239        unit_of_time:
240
241          type: string
242          enum:
```

```
243            - minute
244            - hour
245            - day
246            - month
247            - year
248
249     #========================
250     # Record Duplication
251     #========================
252
253     duplicate record of:
254
255       oneOf:
256         - type: array
257           items:
258             $ref: "./01_asset.yml"
259         - type: null
260
261     #========================
262
263     record retirement information:
264       $ref: "./00_common_definitions.yml#/definitions/record retirement definition"
265
266     #========================================
267     #
268     #========================================
269
270     meter_condition_definition:
271
272       properties:
273
274         meter:
275
276           description: is a selection of a pre-defined meter.
277           $ref: default_WMS_meter_object
278
279         numeric_frequency_value:
280
281           oneOf:
282             - type: null
283             - type: number
284
285         characteristic_trigger_value:
286
287           oneOf:
288             - type: null
289             - type: string
290
291
292     reference_attachment:
293
294       properties:
295
```

```yaml
296          document:
297          document_content_class:
298
299            type: string
300            enum:
301              - maintenance manual
302              - asset photograph
303              -
304
305    #=======================================
306    # Compliance Requirement Definition
307    #=======================================
308
309    compliance_requirement:
310
311      properties:
312
313        name:
314
315          type: string
316          $comment: |
317            For example, 'ANSI Z358.1-2014 on weekly inspection of self-contained
     ↪  emergency wash equipment'.
318
319        requirement_detail:
320
321          description: provides all relevant descriptions
322          type: string
323          $comment: |
324            The following is an example of the requirement detail text for a
     ↪  compliance requirement.
325
326                applicable_asset_class:
327                  - emergency eye-wash
328
329                requirement_source(s):
330                  - ANSI Z358.1-2014 / Emergency Eyewash & Shower Standard / 4 Emergency
     ↪  Showers / 4.6 Maintenance and Training
331                     ...
332                  - ANSI Z358.1-2014 / Emergency Eyewash & Shower Standard / 4 Emergency
     ↪  Showers / 4.5 Installation
333                     ...
334
335                source_content_guide:
336                  - ANSI Z358.1-2014 4.6 states the requirement to check that shower
     ↪  still meets standards
337                  - ANSI Z358.1-2014 4.5 states the standards to apply for ht check
338
339                perform_every:
340                  - year
341
342          requirement_compliance_class:
343
```

```
344          description: indicates the level of compliance, with legislative being the
     ↪   top
345          $ref: "#/compliance_class"
346
347   compliance_class:
348
349      type: string
350      enum:
351        - legislative
352        - corporate policy
353
354
355    #############################################
356    # Common Work Entity Definition
357    #############################################
```

01_asset.yml

```
1    ---
2    $schema: "http://json-schema.org/draft-07/schema#"
3    title: asset
4    $id:
     ↪   https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
5    type: object
6
7    $comment: >
8
9    properties:
10
11     ID:
12
13       type: string
14       description: A read-only UUID, generated by the system, to uniquely identify the
     ↪   asset record.
15       rule_spec:
16         - name: vertical asset ID
17           spec_ID: 41JeoQuvex
18           type: [assertion]
19           specification: |
20             Upon the creation of a new vertical facility asset record generate a
     ↪   unique ID (such as UUID Ver4)
21           checked_on: 2024-08-15
22           $comment: |
23             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.
24         - name: linear_asset_ID
25           spec_ID: Vku-67dDxx
26           type: [assertion]
27           specification: |
```

```yaml
28            Upon the creation of a new asset record corresponding to a record in TWAG,
   ↪   through the Maximo-TWAG integration,
29                populate the TWAG_asset record's "Facility ID" value into the "ID".
30          checked_on: 2024-08-15
31          $comment: see comment for rule 41JeoQuvex.
32
33    name:
34
35      type: string
36      description: The human readable short description of the asset.
37      $comment: |
38          Assumption: an non-is a specific commercial product is always built on site
   ↪   for a specific purpose, and would permanently occupy a role. An example is an
   ↪   aeration tank.
39      rule_spec:
40        - name: Asset Naming
41          spec_ID: 4ykhOm_Dle
42          type: assertion
43          specification: |
44            if asset_x.properties."is_a_commercially_available_product" = TRUE
45              asset_x.properties.name is the semi-colon delimited concatenation of:
46                - asset_x.properties.class.properties."class name"
47                - asset_x.properties."item_product_master_record".properties.product
   ↪   manufacturer company
48                -
   ↪   asset_x.properties."item_product_master_record".properties.model_and_sub-model
49                - asset_x.properties."item_product_master_record".properties.product
   ↪   configuration code
50                - asset_x.properties."OEM_serial"
51            elif asset_x.properties."is_a_commercially_available_product" = FALSE
52              asset_x.properties.name is the semi-colon delimited concatenation of:
53                - asset_x.properties.class.properties."class name"
54                - asset_x.properties."assigned_to_role".properties.name
55            # NOTE: actual script should contain additional condition handle
   ↪   formatting of the name text when there is missing data in any concatenated
   ↪   property.
56          status:
57            checked: 2024-08-15
58
59    class:
60
61      $ref: "../B-entity_class_object_schema/01_asset_item_tool_class.yml"
62      description: Indicates the primitive class to which this asset is an instance.
63      rule_spec:
64        - name: exclusion of parts ("only used as a part asset") from asset
   ↪   classification
65          spec_ID: V15NNHZuxl
66          type: [validation, UI]
67          specification: |
68            Assertion Part:
69              For all assets "asset_x",
70                the value of (asset_x.class.properties.only used as a part asset) must
   ↪   be FALSE
```

```yaml
71             UI Part:
72                In all asset classification search or selection screens, eliminate or
   ↪  filter out all classes "class_y",

73
74                  where (class_y.properties.only used as a part asset) is TRUE
75           checked_on: 2024-08-15
76
77   class_dependent_specifications:
78
79       type: object
80       description: is a set of properties applicable to the class.
81
82   inferred_classes:
83
84       oneOf:
85         - type: array
86           items:
87             type: string
88         - type: null
89       read-only: true
90       integration: true
91       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.
92       $comment: |
93         To implementer: this field will be populated by a rule processor, operating
   ↪  outside of Maximo and with integration to Maximo. An example of the inferred
   ↪  class is "high-pressure boiler".  The values are strings instead of
   ↪  classification objects because the inferred classes will not be in Maximo's
   ↪  asset classification.
94
95
96  ## STATE AND STATUS GROUP OF PROPERTIES
97     #====================================
98
99   physical_status:
100
101      type: string
102      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).
103      enum:
104        - planned
105        - in possession
106        - installed
107        - abandoned in place
108        - removed from possession
109        - missing
110        - lost
111      $comment: |
112        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".
```

```yaml
113
114    operating_state:
115
116      type: string
117      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.
118      enum:
119        - available (up)
120        - unavailable (down)
121        - not applicable
122      $comment: |
123        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, organization, or anyone.
124
125 ## OWNER, OPERATOR, MAINTAINER GROUP
126    #=================================
127
128    owned_by_organization:
129
130      oneOf: [type: null, $ref: "./04_org_or_group.yml"]
131      description: Denotes the organization that owns the asset.
132      integration: true
133
134      rule_spec:
135        - name: valid values of "owned_by_organization" property
136          spec_ID: 410N2dr_xx
137          type: [validation,UI]
138          specification: |
139            - The valid range of values for selection includes the first or second of
    ↪  the organizational hierarchy, specified in the
    ↪  (\TWmaximoConfig\3-System_Hierarchies\04_org_hierarchy.md) . For examples,
140              - first level example: York Region,
141              - second level example: Toronto Water
142            - The UI must only present the valid range of values to the users for
    ↪  selection, and the valid range of values must be presented as a hierarchy.
143          checked_on: 2024-08-19
144
145    owned_by_another_organization:
146
147      oneOf:
148      - type: null
149      - type: string
150      description: name of an organization that is not found in the current list (and
    ↪  should be added)
151      integration: true
152      $comment: |
153        []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 organization that owns it had not been
    ↪  added to the value list.
154
```

```
155    maintenance_group:
156
157      oneOf: [type: null, $ref: "./04_org_or_group.yml"]
158      description: group responsible for the overall maintenance of the asset - for
     ↪   example, a unit, work area, or crew.
159
160    operator_group:
161
162      oneOf: [$ref: "./04_org_or_group.yml"]
163      description: group responsible for the operation of the asset
164      $comment: |
165        This property is not strictly needed for the vertical assets - their group can
     ↪   be inferred from their assignment (to role or user) values.  This value is need
     ↪   for TWAG / linear assets.
166
167      rule_spec:
168        - name: inheriting the asset's maintenance and operator group values from its
     ↪   role
169          spec_ID: VJ1QRgIclg
170          specification: |
171            - if the value of asset_x.properties.assigned_to_role is role_y, then
172                inherit the value of
173                  - asset_x.properties.maintenance_group
174                  - asset_x.properties.operator_group
175                from the same properties of role_y
176          checked_on: 2024-08-20
177
178
179  ## ASSIGNMENT GROUP OF PROPERTIES
180    #===============================
181
182    assignment_type:
183
184      oneOf:
185        - type: string
186        - type: null
187      description:
188      enum:
189        - to a role
190        - to a user group
191        - to a single user
192        - not assigned
193      $comment: |
194        This property is added to assist with the interpretation of the null value in
     ↪   the "assigned_to_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.
195
196      rule_spec:
197        - name: Rendering of (asset_x.properties."assignment_type") data field.
198          spec_ID: 4yARRuvOex
199          type: UI
```

```
200          description: |
201            the options of this property should be presented as radial button
202          status: specified
203
204      - name: Valid Assignment of an Asset
205        spec_ID: NyrzGKwuel
206        type: [validation, assertion, UI]
207        description: |
208          If asset_x.properties."assignment_type" = "to a role", then
209            - asset_x.properties."assigned_to_role" must NOT = null;
210            - asset_x.properties."assigned_to_tool_user" must = null
211            - asset_x.properties."assigned_to_tool_user_group" must = null
212          elif .properties."assignment_type" = "to a user group", then
213            - asset_x.properties."assigned_to_tool_user_group" must NOT = null;
214            - asset_x.properties."assigned_to_role" must = null
215            - asset_x.properties."assigned_to_tool_user" must = null
216          elif asset_x.properties."assignment_type" = "to a single user", then
217            - asset_x.properties."assigned_to_tool_user" must NOT = null;
218            - asset_x.properties."assigned_to_role" must = null
219            - asset_x.properties."assigned_to_tool_user_group" must = null
220          elif asset_x.properties."assignment_type" = null, then
221            - asset_x.properties."assigned_to_role" must = null
222            - asset_x.properties."assigned_to_tool_user" must = null
223            - asset_x.properties."assigned_to_tool_user_group" must = null
224          Also, in the UI screen, disable the properties that should = null
225        status: specified
226
227  assigned_to_role:
228
229    oneOf:
230      - $ref: "./02_role.yml"
231      - type: null
232    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).
233
234  assigned_to_tool_user_group:
235
236    oneOf:
237      - $ref: "./04_org_or_group.yml"
238      - type: null
239    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.
240
241  assigned_to_tool_user:
242
243    oneOf:
244      - $ref: "./02_role.yml"
245      - type: null
246    description: Indicates the assignment of an asset (usually a tool) that does not
   ↪  have a system role.
247
```

```yaml
248
249  ## LOCATION PROPERTY GROUP
250    #========================
251
252    installation_or_parking_location:
253
254      oneOf:
255        - $ref: "./03_space.yml"
256        - type: null
257
258    service_address_or_coordinate:
259
260      oneOf:
261        - $ref: MaximoServiceAddressObject
262        - type: null
263      $comment: |
264        this is referencing Maximo's native service address object
265      rule_spec:
266          - name: asset present at site must have location information on record
267            spec_ID: 01J5R2F9ARJDM3RMGE9WYZWVFE
268            type: [validation]
269            specification: |
270              if the value of asset_x.properties.physical_status is either
271                - "in possession", or
272                - "installed"
273              then at least one of the following properties must NOT be null
274                - asset_x.properties.installation_or_parking_location
275                - asset_x.properties.service_address_or_coordinate
276            check_on: 2024-08-20
277
278    parent_asset:
279
280      oneOf:
281        - $ref: "./01_asset.yml"
282        - type: null
283      description: >
284        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.
285      $comment: |
286        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.
287
288
289  ## PRODUCT AND TOOL ASSOCIATION GROUP
290    #=================================
291
```

```
292    is_a_commercially_available_product:
293
294      type: boolean
295      description: An asset is made under as a product of a commercial entity, as
    ↪  opposed to an asset that is assembled on site.
296      $comment: No null value allowed because this information is self-evident
297
298      rule_spec:
299
300        - name: Automatic Value Assignment to
    ↪  properties."is_a_commercially_available_product"
301          spec_ID: 4Jg2gYSOee
302          type: [assertion]
303          specification: |
304            - Upon record creation, set the value to TRUE.
305            - Upon a asset_x.properties.class value change or a re-run of the Maximo
    ↪  rule processor,
306                if asset_x.properties.class.properties."non-manufactured" = TRUE;
307                  set the value to TRUE;
308                else set the value to FALSE.
309          status: [specified]
310
311    is_a_tool:
312
313      type: boolean
314      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).
315      $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.
316
317      rule_spec:
318
319        - name: Value of (.properties."is_a_tool") defaults to false
320          spec_ID: 41sz7KSdxe
321          type: assertion
322          specification: |
323            - Upon record creation, set the default value to FALSE.
324            - Upon a asset_x.properties.class value change or a re-run of the Maximo
    ↪  rule processor,
325                if asset_x.properties.class.properties.tool = TRUE;
326                  then set the value to TRUE;
327          status: [specified, checked]
328
329    is_mobile:
330
331      oneOf:
332        - type: boolean
333        - type: null
334
```

```yaml
335      rule_spec:

336

337        - name: mobiles need to have its serial number on record
338          spec_ID: EyA3sYa9le
339          type: validation
340          specification: |
341            For any asset_x,
342              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
343                the value of asset_x.properties.OEM_serial cannot be null.
344          check_on: 2024-08-20

345

346    item_product_master_record:

347

348      oneOf:
349        - $ref: "./05_item_master.yml"
350        - type: null
351      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.

352

353      rule_spec:

354

355        # - name: If an asset is commercially available but not a tool, then it must
     ↪ have mfr and model information.
356        #   spec_ID: VJY43yI9lx
357        #   type: [assertion, UI]
358        #   specification: |
359        #     if asset_x.properties.is_a_commercially_available_product = TRUE AND
     ↪ asset_x.properties.is_a_tool = FALSE, then
360        #         - (asset_x.properties."item_product_master_record") is NOT null
361        #         - enable (asset_x.properties."item_product_master_record") in UI
362        #     else
363        #         - (asset_x.properties."item_product_master_record") is null
364        #         - disable (asset_x.properties."item_product_master_record") in UI
365        #   status: TBS

366

367

368        - name: valid item master record in .properties.item_product_master_record
369          spec_ID: VJGKn1I9ex
370          type: validation
371          specification: |
372            For asset_x.properties.item_product_master_record,
373              only accept a master record whose value of
     ↪ asset_x.properties.generic_or_specific_product is "specific commercial product".
     ↪
374          status: [specified, checked]
375          $comment: related to 4y3dRfLcee

376

377    tool_product_master_record:

378

379      oneOf:
380        - $ref: "./06_tool_master.yml"
```

```yaml
381        - type: null
382      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.
383
384      rule_spec:
385
386        - name: when to enable the tool_product_master_record
387          spec_ID: NyQBbeL9xl
388          specification: |
389            if asset_x.properties."is_a_tool" = TRUE
390              then enable (asset_x.properties."tool_product_master_record") property.
   ↪
391          status: [specified, checked]
392
393
394        - name: valid value of asset_x.properties.tool_product_master_record
395          spec_ID: NyFFWlUcll
396          type: validation
397          specification: |
398            only accept a tool_product_master_record whose
   ↪  .properties.generic_or_specific_product property value is "specific commercial
   ↪  product"
399          checked_on: 2024-08-20
400
401
402        - name: an asset may either be associated with a tool or an asset, not both
403          spec_ID:
404          type:
405          specification:
406          status: TBS
407          checked_on:
408
409      TW_workflow:
410
411        - name: creating a rotating tool directly from an asset record
412          specification: TBS []
413          status: work in progress
414
415
416  ## MANUFACTURER AND MODEL GROUP
417     #=============================
418
419    commercial_product_information:
420      oneOf: [type: null,
   ↪  $ref:"./00_common_definitions.yml#/definitions/plain-text_manufacturer_and_model_def"]
421
422  ## DATE PROPERTY GROUP
423     #========================
424
425    construction_contract_number:
426
```

```yaml
427       oneOf:
428         - type: string
429         - type: null
430       description: The construction_contract_number (usually RFQ#) assigned by the
    ↪  City
431
432    first_day_of_City_operation:
433
434       oneOf:
435         - type: string
436         - type: null
437       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.
438       $comment: |
439         This usually coincides with "warranty start date".  However, if the asset is
    ↪  not delivered through a project, "warranty start date" may be empty.
440
441    OEM_serial:
442
443       oneOf:
444         - type: string
445         - type: null
446       description: The serial number, affixed on the asset, designated by the
    ↪  manufacturer.
447       $comment: |
448         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).
449
450    purchase_cost_in_CAD:
451
452       description: the original purchase cost of the asset (not necessary if the asset
    ↪  is associated with a item master record)
453       oneOf:
454         - type: number
455         - type: null
456
457    asset_photos:
458
459       oneOf:
460         - type: array # "array" indicates asset may have multiple photos
461           items:
462             - type: string # photos are converted to a string in JSON
463             - oneOf:
464               - contentMediaType: image/png
465               - contentMediaType: image/jpg
466         - type: null
467
468
469  ## BACKGROUND PROPERTIES POPULATED AUTOMATICALLY
```

```yaml
470    #===========================================

471

472    TW Asset Group:

473

474      oneOf:
475        - type: string
476        - type: null
477      invisible: true
478      read-only: true
479      enum:
480        - Drinking Water Network
481        - Drinking Water Treatment Plants
482        - Waste and Storm Water Network
483        - Wastewater Treatment Plants
484        - Independent Building
485        - Multiple Major Systems
486      $comment: |
487        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.

488
489    ##############################################################
490    # HIGH LEVEL RULES
491    ##############################################################

492
493    rule_spec:

494
495      - name: Asset must have a start of operation date info before we can indicate that
    ↪  it is operationally available.
496      spec_ID: NyG2nzL5xg
497      type: validation
498      specification: |
499        if both of the following properties are null
500            - (asset_x.properties."first date of City operation")
501            - (asset_x.properties."warranty start date")
502          then the value of (asset_x.properties."operating_state") CANNOT be
    ↪  "available (up)"

503

504

505      - name: asset can be assigned exclusively to either a role, user, or user group
506      spec_ID: EkD-ZmIceg
507      type: validation
508      specification: |
509        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)
510            - asset_x.properties."assigned_to_role"
511            - asset_x.properties."assigned_to_tool_user"
512            - asset_x.properties."assigned_to_tool_user_group"
513      errorMessage: "Between \"assigned_to_role\", \"assigned_to_tool_user_group\",
    ↪  \"installed\"assigned_to_tool_user\", every asset may only have one type
    ↪  assignment."

514
```

```yml
515
516    - name: consistency between (.properties."operating_state") and assignment values
517      spec_ID: 41OFxr8ceg
518      type: [validation, assertion]
519      specification: |
520        if an asset does not have a value in any of the following properties (i.e.,
      ↪ all nulls),
521            - asset_x.properties."assigned_to_role"
522            - asset_x.properties."assigned_to_tool_user"
523            - asset_x.properties."assigned_to_tool_user_group"
524          then the value of (asset_x.properties."operating_state") must be "not
      ↪ assigned work". The opposite must also be true.
525      errorMessage: "An asset NOT assigned to a role, user, or user group should not
      ↪ be operating and therefore would not have an operating_state"
526
527
528    - name: consistency between asset's physical status and its operating state and
      ↪ assignments
529      spec_ID: NyG2nzL5xg
530      type: [assertion, validation]
531      specification: |
532        If the value of (asset_x.properties."physical_status") is NEITHER of the
      ↪ following
533            - "installed"
534            - "in possession"
535          then the following properties would take on the stated values
536            asset_x.properties."operating_state" = "not applicable"
537            asset_x.properties."assigned_to_role" = null
538            asset_x.properties."assigned_to_tool_user" = null
539            asset_x.properties."assigned_to_tool_user_group" = null
540      errorMessage: "If an asset is not \"installed \\ in possession\" (see the
      ↪ physical_status), then it should not have an operating_state value or any
      ↪ assignment."
541
542
543    - name: an asset can only be assigned to a discrete asset role
544      spec_ID: 4yBXuH8qle
545      type: [validation]
546      specification: |
547        if (asset_x.properties."assigned_to_role") is NOT null
548            then
      ↪ (asset_x.properties."assigned_to_role".properties.class.properties."discrete
      ↪ asset role") = TRUE
549      errorMessage: an asset can only be assigned to a discrete asset role
550
551
552    - name: inheriting the asset location information from its role
553      spec_ID: NJdGTHLqeg
554      type: [assertion]
555      specification: |
556          For an asset, asset_x, if
557            all of the following are true:
558              - asset_x.properties."operating_state" = "installed"
```

```
559              - asset_x.properties."assigned_to_role" is NOT null
560         and one of the following is true
561              - asset_x.properties."assigned_to_role".properties."asset installation
↪   location" is NOT null
562              -
↪   asset_x.properties."assigned_to_role".properties."service_address_or_coordinate"
↪   is NOT null
563         then
564              (asset_x.properties."installation_or_parking_location") would be set to
↪   the value of
565              (asset_x.properties."assigned_to_role".properties."asset installation
↪   location")
566              (asset_x.properties."service_address_or_coordinate") would be set to the
↪   value of
567
↪   (asset_x.properties."assigned_to_role".properties."service_address_or_coordinate")
568
569
570   - name: over-write of manufacturer and model information
571     spec_ID: 01J5RPPEKJCP11NBRW3A4XCKF7
572     specification: |
573       if the value of either
574     status: TBS
575
576
577   - name: a commercially available asset must be associated with manufacturer and
↪   model information
578     spec_ID: 4y3dRfLcee
579     type: validation
580     exempt_grandfather: true
581     specification: |
582       if the value of asset_x.properties.is_a_commercially_available_product is TRUE
583         then NONE of the following properties can be null
584           - asset_x.properties.item_product_master_record
585           - asset_x.properties.manufacturer_name
586           - asset_x.properties.product_model_information
```

02_role.yml

```
1   ---
2   $schema: "http://json-schema.org/draft-07/schema#"
3   title: role
4   $id:
↪   https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
5   type: object
6
7   properties:
8
9     ID:
10
11       type: string
```

```yml
12      description: Also known as the "tag number" or "entity number" in Avantis's
   ↪  vocabulary. (Avantis is the a WMS).
13      #[] to do ID for pumping stations an chambers will be from the GIS.
14      rule_spec:
15        - name: ID of Linear Assets Represented as Role in Maximo
16          spec_ID: Vku-67dDxx
17          type: triggered action
18          specification:
19            trigger: replication creation of assetY record from the TWAG
20            action: apply Facility ID from TWAG as ID
21          status: false
22
23  name:
24
25      type: string
26      description: A structured and ideally unique description of the role.
27      const: > #[]
28        ${properties.class.properties."class name"};
29        childOf: ${properties."parent entity".properties."role number"};
30        serving: ${properties."client role served".properties."role number"};
31      # MT []: would you like me to move this into individual roles?
32
33  parent:
34
35      $ref: "./02_role.yml"
36      description: References the role that is served by the larger asset, which
   ↪  physically subsumes the asset serving this role.
37
38  class:
39
40      $ref: B-entity_class_object_schema/02_role_class.yml
41      description: The class denote the broad types of asset that may play the role
   ↪  (e.g., "breaker role"), and sometimes, more specifically, it also denotes the
   ↪  useful function provided by that asset in the role to the larger system (e.g.,
   ↪  the "tie-breaker role" provides tie-breaking function to the electrical
   ↪  distribution system).
42
43  class_dependent_specifications:
44
45      type: object
46      description: is a set of properties applicable to the class.
47
48  inferred_classes:
49
50      oneOf:
51        - type: array
52          items:
53            type: string
54        - type: null
55      items:
56        type: string
57      read-only: true
58      description: See the description for inferred classe names(s) made in the
   ↪  \TWmaximoConfig\1-Schemas\A-entity_record_schema/01_asset_schema.yml
```

```yaml
59        $comment: |
60          See the comment for inferred classe names(s) made in the
   ↪    \TWmaximoConfig\1-Schemas\A-entity_record_schema/01_asset_schema.yml
61
62    role_status:
63
64        type: string
65        enum:
66          - specified
67          - active
68          - eliminated
69
70        description: |
71          This is the basic life-cycle status of a role. 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.
72
73        rule_spec: |
74          #[]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.
75            $comment: a procedure should to be created to allow the recursive
   ↪    elimination of a role and all of its children.
76          #[]RULE VygDC0Frxl: 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.)
77            []Review with To ASMP: with this rule, we no longer need to have a hierarchy
   ↪    branched for retired roles.
78          #[]RULE EkP5qy5Sxl: If change auditing cannot be turned on, then when a role
   ↪    record status is "eliminated", all the specification in the record's datafields
   ↪    must be frozen.
79
80    role technical requirement description:
81
82        type: string
83        description: A free text description of the role's performance requirements that
   ↪    must be satisfied by the asset. For example, the lifting capacity in kilograms
   ↪    for a crane.
84        $comment: |
85          Eventually, this information should be replaced by formalized properties
   ↪    associated with specific classes. For example, for the motor role class, there
   ↪    would be a requirement for a minimum amount of horsepower or torque.
86
87    serving_asset_in_role:
88
89        $ref: "./02_role.yml"
90        description: Identifies the asset this role is serving. For example, given a
   ↪    motor starter role, the value in this data field identifies the role of the
   ↪    motor controlled by that motor starter.
91
92    asset installation location:
93        oneOf:
```

```yaml
 94          - $ref: "./03_space.yml"
 95          - type: null
 96        description: References the space in which the asset serving the role would be
    ↳   installed.
 97
 98      service_address_or_coordinate:
 99        oneOf: [$ref: MaximoServiceAddressObject, type: null]
100        description: A geo-coordinate or the nearest street address of the asset.
101
102      #========================
103
104      operator_group:
105
106      inherit operator group value:
107        type: boolean
108        default value: true
109
110      #========================
111
112      inherit_operator_group_from_parent:
113
114        type: boolean
115        default_value: true
116
117      #========================
118
119      maintenance_group:
120
121        oneOf: [$ref: "./04_org_or_group.yml", type: null]
122        description: Group responsible for the preventive and reactive maintenance of
    ↳   the asset in the role. For example, a unit, work area, or crew.
123
124      #========================
125
126      inherit_maintenance_group_from_parent:
127
128        type: boolean
129        default_value: true
130
131      #========================
132
133      operational criticality:
134
135        oneOf: [$ref:'#/definitions/criticality rating definition', type: null] # see
    ↳   section 3. LOCAL OBJECT DEFINITION
136        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.
137
138      #========================
139
140      protective function criticality:
141
```

```
142      oneOf: [$ref:'#/definitions/criticalityRatingDef', type: null] # see section 3.
     ↪  LOCAL OBJECT DEFINITION
143      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.
144
145    #========================
146
147    duplicate record of:
148      oneOf:
149        - type: array
150          items:
151            $ref: "./02_role.yml"
152        - type: null
153
154    #========================
155
156    record retirement information:
157      $ref: "./00_common_definitions.yml#/definitions/record retirement definition"
158
159
160    #========================
161    # [] REMOVE THE FOLLOWING AFTER RULE ABOUT ID IS REVISED
162    # GIS object ID:
163    #   type: string
164    #   description: This is the ID of the equivalent object (asset or role) in GIS.
     ↪  This value suggests the original record was created in GIS and copied to WMS
     ↪  through the data integration link.
165
166
167
168  ############################################################
169  # 2. DATA INTEGRITY RULES
170  ############################################################
171    allOf:
172      - if:
173          oneOf:
174            - roleClass:
175                properties:
176                    className:
177                      const: pumping station
178            - roleClass:
179                properties:
180                    className:
181                      const: water treatment plant
182            - roleClass:
183                properties:
184                    className:
185                      const: large chamber
186        then:
187          required: GIS object ID
188
```

```yaml
189    #[]RULE:
190    # 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_role" property).
191
192  ###############################################################
193  # 3. LOCAL OBJECT DEFINITION
194  ###############################################################
195
196  definitions:
197    criticality rating definition:
198      type: object
199      properties:
200        rating:
201          type: integer
202        description:
203          type: string
204      enum:
205        - rating: 1
206          description: TBD
207        - rating: 2
208          description: TBD
209        - rating: 3
210          description: TBD
211        - rating: 4
212          description: TBD
213        - rating: 5
214          description: TBD
215
216  $comment: |
217    OPEN AND TODO ITEMS:
218      [x] To have discussion on what we put in for criticality
219      [] explicitly specify an unoccupied role
220      [] expression of role equivalence
221      [] RULE: certain GIS assets, such as pumping stations, or treatment facilities
    ↪   must be mapped over as roles
```

03_space.yml

```yaml
1   ---
2   $schema: "http://json-schema.org/draft-07/schema#"
3   title: space
4   $id:
    ↪   https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
5   type: object
6
7   properties:
8
9     ID:
10
11      type: string
```

```yaml
12      description: An unique ID
13      $comment: In the future, this value should be validated with a regular
   ↪  expression.

15   parent:

17      $ref: "./03_space.yml"
18      description: The larger space that completely contains this space.

20   name:

22      type: string
23      description: Short name. Should be one that is commonly use by staff in
   ↪  communication. For example, "boardroom"

25   complete_name:

27      type: string
28      read-only: true
29      rule_spec:
30        - spec_ID:
31          name: complete name generation
32          id: 01JFVNS0YDFD7K5DP4NVMSKTY8
33          status:
34      $comment: |
35        Automatically generated by the system and not editable. The value is name to
   ↪  that of its parent, its grandparent ... all the way up that facility space.


38   enclosed_by_asset:

40      oneOf: [$ref: "./01_asset.yml", type: null]
41      description: indicates that the space is what is enclosed by (and immediately
   ↪  surrounding) the asset, such as a
42        - building,
43        - structural tank,
44        - equipment cabinet,
45        - vehicle

47   class:

49      $ref: spaceClassObject
50      $comment: |
51        see space_classification.md file

53   class_dependent_specifications:

55      type: object
56      description: is a set of properties applicable to the class.

58   inferred_classes:

60      oneOf:
```

```yaml
61        - type: array
62          items:
63            type: string
64        - type: null
65      read-only: TRUE
66      $comment: |
67        See the comment for inferred class name(s) made in the
   ↪  \TWmaximoConfig\1-Schemas\A-entity_record_schema/01_asset_schema.yml
68
69        #[]REQ 41Vru1Rrxe: This data field should be visible to the users, but should
   ↪  not be editable by the user
70
71  service_address_or_coordinate:
72      oneOf: [$ref: .IBM_Maximo_object/MaximoServiceAddressObject, type: null]
73      $comment: this is referencing Maximo's native service address object
74
75    status:
76
77      type: string
78      enum:
79        - specified
80        - realized
81        - eliminated
82      $comment: |
83        This field allows the user to specify whether the space is merely specified,
   ↪  or whether the boundary enclosing the space have been contructed (or
   ↪  alternatively, the fiat property / or area boundary around is formally
   ↪  established and approved.) - i.e. "exists".
84        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.
85        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.
86
87    confined_space:
88
89      type: boolean
90
91    inherit_hazardous_property_values:
92
93      type: boolean
94      default_value: true
95
96    hazardous location class:
97
98      type: string
99      enum:
100        - I
101        - II
102        - III
103
104    hazardous location division:
```

```
105
106        type: string
107        enum:
108          - 1
109          - 2
110
111    hazardous location group:
112
113        type: string
114        enum:
115          - A
116          - B
117          - C
118          - D
119          - E
120          - F
121          - G
122
123
124    duplicate record of:
125
126        oneOf:
127          - type: array
128            items:
129              $ref: "./03_space.yml"
130          - type: null
131
132    record retirement information:
133        $ref: "./00_common_definitions.yml#/definitions/record retirement definition"
134
```

04_org_objects_definitions.yml

```
1   ---
2   $schema: "http://json-schema.org/draft-07/schema#"
3   title: organization
4   $id:
    ↪   https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
5   type: object
6   properties:
7
8     organization_or_group_name:
9
10       type: string
11
12     parent_organization:
13
14       oneOf:
15         - $ref: "./04_org_or_group.yml"
16         - type: null
17
```

```yaml
18    class:
19
20      $ref: B-entity_class_object_schema/04_org_class.yml
21
22    class_dependent_specifications:
23
24      type: object
25      description: is a set of properties applicable to the class.
26
27    leader:
28
29      oneOf:
30        - $ref: "./07_person.yml"
31        - type: null
32      description:
33
34    equivalent_to_Maximo_site:
35
36      description: indicates that this organization maps to a particular site (a
↪        native Maximo object)
37      $ref: MaximoSiteObject
38
39    equivalent_to_Maximo_org:
40
41      description: indicates that this organization maps to a particular organization
↪        (a native Maximo object)
42      $ref: MaximoOrgObject
43
44    duplicate record of:
45
46      oneOf:
47        - type: array
48          items:
49            $ref: "./04_org_or_group.yml"
50        - type: null
51
52    record retirement information:
53
54      $ref: "./00_common_definitions.yml#/definitions/record retirement definition"
55
56 supervisor_group:
57
58 crew:
59
60
```

05_item_master.yml

```yaml
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: item master
```

```yaml
$id:
  ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
type: object

properties:

  number:

    type: string
    description:
      A read-only UUID, generated by the system, to uniquely identify the item.
    $comment: |
      UUID instead of a simple serial used with the consideration that we may in the
  ↪  future incorporate items defined outside of TW.

  #========================

  name:

    type: string
    description: The human readable short description of the item.

    rule_spec:

      - name: item master record naming
        form: long
        spec_ID: VkYgCtRPlx
        type: assertion
        specification: |
          if (item_x.properties.commodity_or_commercial_product) = "commodity", then
  ↪
            the value of item_x.properties.name would be the semi-colon delimited
  ↪  concatenation of the following property values:
              - properties.class.properties.name
              - every non-empty class dependent specification values
              - properties.supplementary_commodity_description
          elif: (item_x.properties.commodity_or_commercial_product = "commercial
  ↪  product"), then:
            the value of item_x.properties.name would be the semi-colon delimited
  ↪  concatenation of the following property values:
              - properties.class.properties.name
              - properties.product manufacturer company.properties."company name"
              - properties.model_and_sub-model
              - properties.version_or_model_year
              - properties.product configuration code
        checked on: 2024-08-15

  #========================

  class:

    $ref: "../B-entity_class_object_schema/01_asset_item_tool_class.yml"
    $comment: is a value from the item classification, which is a superset of the
  ↪  asset class.
```

```yaml
51
52      rule_spec:
53
54        - name: item classification list includes all classes
55          spec_ID: EynXVZ-dxg
56          specification: |
57            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
58              - properties.tool
59              - properties.only used as a part asset
60          status: specified
61          checked_on: 2024-08-15
62
63
64    class_dependent_specifications:
65
66      type: object
67      description: is a set of properties applicable to the class value.
68
69  # INVENTORY MANAGEMENT FLAGS
70    #=========================
71
72    stocked_at_TW:
73
74      $ref: "./00_common_definitions.yml#/definitions/stocked_at_TW_def"
75
76    rotating:
77
78      $ref: "./00_common_definitions.yml#/definitions/rotating_property_def"
79
80    #====================================
81    # generic and specific definition
82    #====================================
83
84    commodity_or_commercial_product:
85
86      type: string
87      description: indicate whether the item master defines an unspecialized commodity
    ↪ or a specific commercial product.
88      enum:
89        - commodity
90        - commercial product
91
92    supplementary_commodity_description:
93
94      type: string
95      description: supplementary description, in addition to the class value and and
    ↪ class-dependent specification values, necessary to differentiate a commodity.
96
97    #====================================
98    # MANUFACTURER AND MODEL GROUP
99    #====================================
```

```yaml
100
101    commercial_product:
102
103      oneOf:
104        - type: null
105        - $ref: "./00_common_definitions.yml#/definitions/manufacturer_and_model_def"
106
107    commercial_product_description:
108
109      oneOf:
110      - type: null
111      - type: string
112        $ref:
    ↪  "./00_common_definitions.yml#/definitions/plain-text_manufacturer_and_model_def"
113
114    #========================
115
116    instant_of_commodities:
117
118      oneOf:
119        - type: null
120        - $ref: "./05_item_master.yml"
121
122      rule_spec:
123        - description: range of the instant_of_commodities property must be  items
    ↪  whose TW_defined_commodity value is 'true'.
124          form: short
125          spec_ID: 01JF33GVB2DT7K5FJ3SA3P9EP4
126
127    #========================
128
129    same_item_as:
130
131      oneOf:
132        - type: array
133          items:
134            $ref: "./04_item_master.yml"
135        - type: null
136      description: |
137        Identifies the same commercial product item made by the same manufacturer, but
    ↪  differing only in item unit format. For example, the 208-litre drum item and the
    ↪  5-litre bottle item of Penzoil 5W30 Synthetic Lubricant.
138
139      rule_spec:
140
141        - name: Infer Symmetrical Equivalence between Product Items
142          spec_ID: EJCnUukuex
143          type: inference
144          specification: ""
145          status: []
146
147        - name: Range must be a commercial product as well
148          spec_ID: 01JF81079K178X9B4NSG23AA0Z
```

```yaml
149          type: validation
150          specification:
151          status: []
152
153    #=========================
154
155    ordering_information:
156
157      oneOf:
158        - type: array
159          items:
160            $ref: "#/definitions/vendor_order_detail"
161        - type: null
162
163    #=========================
164
165    issue_unit:
166
167      type: string
168      description: A description of each individual unit issued for work, such as a
    ↪  can or roll, of the product item, regardless of the ordering unit, such as a box
    ↪  of 24.  For example, "80 ml can" or "27 x 500in. roll".
169      $comment: "To implementer: use the out of the box list from Maximo"
170
171  #=========================
172  # LOCAL OBJECT DEFINITIONS
173  #=========================
174
175  definitions:
176
177    vendor_order_detail:
178
179      properties:
180
181        vendor:
182
183          $ref: MaximoCompanyObject
184
185        #=========================
186
187        order_unit:
188
189          oneOf:
190            - type: null
191            - type: string
192          description: A description of each individual unit of order, such as a can
    ↪  or roll, of the product item, regardless of the ordering unit, such as a box of
    ↪  24.  For example, "80 ml can" or "27 x 500in. roll".
193          $comment: |
194            For a commercial product, if there are multiple item unit formats, one
    ↪  item should be create for each format.
195            To Implementer,
196              Use Maximo's defaults
```

```
197
198        #=======================
199
200        vendor_item_number:
201
202          oneOf:
203            - type: string
204            - type: null
205
206        #=======================
207
208        unit_cost_in_CAD:
209
210          type: number
211          description: The expected cost of a unit of the item. (This should not be
     ↪  the cost of a package of multiple units of the item.)
212          $comment: To TW, in the future, this field should contain a running average
     ↪  of the recent purchase costs, possibly also adjusted for recent inflation.
213
214
215    # 6/25: [] we need to add another field for lead time. This will contribute to the
     ↪  dynamic calculation of criticality among other uses.
```

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/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
5   type: object
6
7   properties:
8
9     number:
10
11      type: string
12      description:
13        A read-only UUID, generated by the system, to uniquely identify the tool.
14      $comment: |
15        UUID instead of a simple serial used with the consideration that we may in the
    ↪  future incorporate items defined outside of TW.
16
17    #=======================
18
19    name:
20
21      type: string
22      description: The human readable short description of the tool.
23      rule_spec:
24        - name: tool master name
```

```yaml
25          spec_ID: VJpSzGxdxg
26          type: implication
27          specification: |
28            if: toolX.properties."tool master type" = "generic tool"
29            then:
30              toolX.properties."tool name" value is the semi-colon ("; ") delimited
   ↪  concatenation of the following property values:
31                - properties.class.properties."class name"
32                - properties."generic tool application definition"
33            elif: toolX.properties."tool type" = "specific commercial product"
34            then:
35              toolX.properties."tool name" value is the semi-colon ("; ") delimited
   ↪  concatenation of the following property values:
36                - properties."tool master class".properties."class name"
37                - properties.product manufacturer company.properties."company name"
38                - properties.model_and_sub-model
39                - properties.version_or_model_year
40                - properties.product configuration code
41          status: specified
42
43    #=======================
44
45    class:
46
47      $ref: "../B-entity_class_object_schema/01_asset_item_tool_class.yml"
48      description: This is a value from the classification, which is a superset of the
   ↪  asset class.
49
50      rule_spec:
51        - name: Tool classification list does not include parts non tools
52          spec_ID: V1ulHHWOgx
53          specification: |
54            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
55          status: specified
56
57    class_dependent_specifications:
58
59      type: object
60      description: is a set of properties applicable to the class.
61
62
63  # INVENTORY MANAGEMENT FLAGS
64    #=======================
65
66    rotating:
67
68      $ref: "./00_common_definitions.yml#/definitions/rotating_property_def"
69
70    #=======================
71
72    mobile:
```

```
73
74     type: boolean
75     description: An tool that is used beyond a permanent installation; instead, it
   ↪   is taken from place to place.
76     $comment: |
77       #PROCESS: SET DEFAULT VALUE:
78         At record creation, set value to false.
79       #PROCESS: EVENT-DRIVEN VALUE CHANGE:
80         Upon the event of a properties.class value change;
81           if properties.class.properties."mobile" = true;
82           then set the value to true;
83           else set the value to false.
84
85
86
87 $comment: ASMP does not expect the tool item master to be widely used during the
   ↪   initial adoption of Maximo - we expect that most tools would initially be
   ↪   represented as un-stocked.
```

07_service_item_master.yml

```
1
```

08_person.yml

```
1  $schema: http://json-schema.org/draft-07/schema#
2  title: Person
3  $id:
   ↪   https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
4  type: object
5
6  properties:
7
8    ID:
9
10     type: string
11     description: Unique identifier for the person
12
13   first_name:
14
15     type: string
16     description: First name of the person
17
18   last_name:
19
20     type: string
21     description: Last name of the person
22
```

```yaml
23    display_name:
24
25      type: string
26      description: Full name displayed (usually a combination of first and last names)
27
28    division:
29
30      type: string
31      description: indicates which City division that the person works for
32
33    unit:
34
35      type: string
36      description: indicates which business-unit that the person works for
37
38    primary_trade:
39
40      type: string
41      description: Primary craft or skill associated with the person
42
43    external_contractor:
44
45      type: boolean
46      description: indicates whether the person is not an employee of the City
47
48    status:
49
50      type: string
51      enum:
52        - ACTIVE
53        - INACTIVE
54      description: Status of the person
55
56    email_address:
57
58      type: string
59      format: email
60      description: Email address of the person
61
62    phone:
63
64      type: string
65      description: Contact phone number of the person
```

09_qualification.yml

```yaml
1  $schema: http://json-schema.org/draft-07/schema#
2  title: Qualification
3  $id:
   ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
4  type: object
```

```yaml
5
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   #========================
14
15   name:
16
17     description: a short name given by the creator of the record.
18     type: string
19
20   #========================
21
22   class:
23
24     type: string
25     enum:
26     - professional license
27     - industry certification
28     - internal certification
29
30   #========================
31
32   issued_by:
33
34     oneOf:
35       - type: null
36       - $ref: "./04_org_or_group.yml"
37
```

10_warranty.yml

```yaml
1  $schema: http://json-schema.org/draft-07/schema#
2  title: Warranty
3  $id:
   ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
4  type: object
5
6  properties:
7
8    #========================
9
10   ID:
11
12     description: is the unique identifier for the warranty contract
13     type: string
14
```

```yml
15    #========================
16
17    description:
18
19      description: is a brief summary of the warranty contract
20      type: string
21
22    #========================
23
24    vendor:
25
26      description: identifies the vendor or provider of the warranty
27      $ref: MaximoCompanyObject
28
29    #========================
30
31    warranty_start_date:
32
33      description: is the first day that the warranty becomes effective
34      oneOf:
35        - type: null
36        - type: number
37
38    #========================
39
40    warranty_expiration_date:
41
42      description: is the last effective date of the warranty
43      oneOf:
44        - type: null
45        - type: number
46
47    #========================
48
49    meter:
50
51      description: if applicable, is the meter reading (e.g., mileage) at which the
   ↪  warranty starts or ends
52      oneOf:
53        - type: number
54        - type: null
55
56    #========================
57
58    covers_labour:
59
60      description: indicates that the vendor is responsible for providing and covering
   ↪  the cost of labour
61      type: boolean
62
63    #========================
64
65    covers_parts:
```

```yaml
66
67     description: indicates that the vendor is responsible for providing and covering
    ↪  the cost of parts
68     type: boolean
69
70   #========================
71
72   specific_terms:
73
74     description: are the terms and conditions related to the warranty coverage
75     type: string
76
77   #========================
78
79   covers_roles:
80
81     description: is the list of roles, more specifically, the assets installed in
    ↪  the roles that are covered by the warranty
82     oneOf:
83       - type: null
84       - type: array
85         items:
86           $ref: "./02_role.yml"
87
88   #========================
89
90   covers_assets:
91
92     description: is the list of assets that are covered by the warranty
93     oneOf:
94       - type: null
95       - type: array
96         items:
97           $ref: "./01_asset.yml"
98     $comment: note that any item converted by the warranty should be expressed as a
    ↪  rotating item (i.e. represented as an asset as well)
99
```

32_job_plan.yml

```yaml
1  ---
2  $schema: http://json-schema.org/draft-07/schema#
3  title: Job Plan
4  $id:
   ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
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:
```

```yaml
10
11    ID:
12
13      description: is a read-only, unique, and permanent ID, generated by the system,
   ↪  to identify the job plan record.
14      type: string
15
16    #================
17
18    name:
19
20      description: is a description of the activity specified in the job plan.
21      type: string
22
23    #================
24
25    discrete_activity_classification:
26
27      description: indicates the type of activity that specified in the job plan
28      $ref: "../B-entity_class_object_schema/32_discrete_activity_class.yml"
29      $comment: this could also be called the activity classification
30
31
32    #==============================
33    # Job Plan Applicability Notes
34    #==============================
35
36    specific_to_asset_classes:
37
38      description: identifies the asset classes on which the activity specified in
   ↪  this job plan can be done.
39      type:
40        oneOf:
41        - type: null
42        - type: array
43          items:
44            $ref: "../B-entity_class_object_schema/01_asset_item_tool_class.yml"
45
46    #================
47
48    specific_to_role_classes:
49
50      description: identifies the role classes - more specifically, the assets
   ↪  installed in these roles - for which this job plan is customized.
51      type:
52        oneOf:
53        - type: null
54        - type: array
55          items:
56            $ref: "../B-entity_class_object_schema/02_role_class.yml"
57      $comment: >
58        Examples of a role class include the tie-breaker and effluent turbidity meter
   ↪  role class.
```

```yaml
59
60    #================
61
62    specific_to_operational_units:
63
64      description: identifies the Toronto Water site(s), defined as a organization in
   ↪  this schema, for which the job plan is specifically customized.
65      type:
66        oneOf:
67        - type: null
68        - type: array
69          items:
70            $ref: "../A-entity_record_schema/04_org_class.yml"
71      $comment: >
72        Examples of operational group include
73          TAB - Ashbridges Bay Wastewater Treatment Plants
74          COL - Waste and Storm Water Collection
75      rule_spec: >
76        - name: job plan's specific_to_operational_units property must be an unit
   ↪  level organization
77          spec_ID: 01JD2V5X97J1Y45JWDW4SV1FJ4
78          type: [validation]
79          specification: |
80            Given an job plan, JP_x, all values of JP_x.specific_to_operational_units
   ↪  must be
81              - an organization, and
82              - whose .class = unit (a subclass of Group in the City)
83
84    #================
85
86    specific_to_roles:
87
88      description: identifies the asset roles for which the job plan is specifically
   ↪  customized.
89      type:
90        oneOf:
91        - type: null
92        - type: array
93          items:
94            $ref: "../A-entity_record_schema/02_role.yml"
95
96    #================
97
98    specific_to_commercial_products:
99
100     description: identifies the the commercial products for which the job plan is
   ↪  specifically written/customized.
101     type:
102       oneOf:
103       - type: null
104       - type: array
105         items:
106           $ref: "../A-entity_record_schema/05_item_master.yml"
```

```yaml
107    rule_spec: >
108      - name: Valid Commercial Product Item reference in a Job Plan
109        spec_ID: 01JD2V5X97J1Y45JWDW4SV1FJ4
110        type: [validation]
111        specification: |
112          Given an job plan, JP_x, all values of
    ↪ JP_x.specific_to_commercial_products must be
113            - a item master record, and
114            - whose .generic_or_specific_product = "specific commercial product"
115
116  #===============================
117  # Record Provenance
118  #===============================
119
120  is_derived_from:
121
122    description: identifies the job plan from which the present job plan
    ↪ specification was based on.
123    oneOf:
124      - type: null
125      - $ref: "./32_job_plan.yml"
126    integration: true
127
128  #================
129
130  failure_codes:
131
132      description: denotes a physical-based failure condition (e.g., shaft
    ↪ misalignment).
133      oneOf:
134        - type: null
135        - type: array
136          items:
137            $ref: "./00_common_definitions.yml#/definitions/failure_code"
138
139  #================
140
141  RCM_failure_modes:
142
143    description: identifies the functional failure mode ID, with respect to a
    ↪ specific asset role, mitigated by the work specified in this job plan.
144    oneOf:
145      - type: null
146      - type: array
147        items:
148          type: string
149    integration: true
150    $comment: >
151      - 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.
152
153  #===============================
154  # Procedure
```

```yaml
155    #===============================
156
157    work_description:
158
159      description: is a single body of text outlining the sequential steps to complete
    ↪    the activity
160      type: string
161      $comment: >
162        Example:
163          1) Ensure you have operational approval before performing this task.
164          2) Follow Lock-out and Tag-out process before starting this task.
165          3) Drain the oil from the gearbox.
166          4) Install 25 Litres of UCON 220 ( food grade) oil.
167          5) Remove Lock-out and Tag-out and check operation.
168          6) Inform operations that the task as assigned is completed.
169
170    # ==========
171
172    requires_shut_down:
173
174      description:
175
176    #========================================
177    # Resource Requirements
178    #========================================
179
180    estimated_duration:
181
182      description: is the estimated time to complete the activity in the job plan
183      $ref: "./00_common_definitions.yml#/definitions/frequency_interval_definition"
184
185    #=================
186
187    parts_or_material_requirements:
188
189      description: identifies the parts and material required to complete the work.
190      oneOf:
191        - type: null
192        - type: array
193          items:
194            $ref:
    ↪    "./00_common_definitions.yml#/definitions/item_requirement_definition"
195
196    #=================
197
198    tool_requirements:
199
200      description: identifies the tools required to complete the work.
201      oneOf:
202        - type: null
203        - type: array
204          items:
205            $ref:
    ↪    "./00_common_definitions.yml#/definitions/tool_requirements_definition"
```

```yml
206
207    #================
208
209    skill_and_trade_requirements:
210
211      description: identifies the trades and qualifications of each trade needed to
    ↪  complete the work.
212      oneOf:
213      - type: null
214      - type: array
215        items:
216          $ref: "./00_common_definitions.yml#/definitions
    ↪  trade_requirement_definition"
217
218    #================
219
220    service_requirements:
221
222      description: identifies (contracted) service needed to complete the work.
223      oneOf:
224        - type: null
225        - type: array
226          items:
227            $ref:
    ↪  "./00_common_definitions.yml#/definitions/service_requirement_definition"
228
229  #======================================
230  # Related Activities
231  #======================================
232
233    must_be_preceded_by:
234
235      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.
236      oneOf:
237      - type: null
238      - type: array
239        items:
240          $ref: "./32_job_plan.yml"
241
242    must_be_followed_by:
243
244      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.
245      oneOf:
246      - type: null
247      - type: array
248        items:
249          $ref: "./32_job_plan.yml"
250
251  #======================================
```

```yaml
252  # Work Trigger Condition Notes
253  #=======================================
254
255    time-based_frequency:
256
257      oneOf:
258        - type: null
259        - $ref:
↪    "./00_common_definitions.yml#/definitions/frequency_interval_definition"
260
261    #================
262
263    meter-based_frequency:
264
265      oneOf:
266        - type: null
267        - $ref: "./00_common_definitions.yml#/definitions/meter_condition_definition"
268
269    #================
270
271    description_of_event-based_trigger:
272
273      oneOf:
274        - type: null
275        - type: object
276          properties:
277
278            relation_to_event:
279
280              type: string
281              enum:
282                - before
283                - during
284                - after
285                - at the start of
286                - at the end of
287
288            description_of_event:
289
290              description: a free-text description of a event or process, such as "an
↪    elevator failure".
291              type: string
292
293    #================
294
295    notes_on_trigger_condition:
296
297      description: free-text description on the additional trigger conditions
298      oneOf:
299        - type: null
300        - type: string
301
302  #=======================================
```

```yaml
303   # Compliance Information
304   #========================================
305
306     compliance_requirement:
307
308       description: identifies the compliance requirement object
309       integration: true
310       oneOf:
311         - type: null
312         - type: array
313           items:
314             $ref: "./00_common_definitions.yml#/definitions/compliance_requirement"
315
316     #=================
317
318     compliance_class:
319
320       description: indicates the level of compliance, with legislative being the top
    ↪   class
321       $ref: "./00_common_definitions.yml#/definitions/compliance_class"
322
323     #=====================
324
325     mitigates_safety_risk_to_staff:
326
327       oneOf:
328       - type: null
329       - type: string
330       enum:
331         - yes
332         - no
333         - unspecified
334
335     #=====================
336
337     mitigates_safety_or_health_risk_to_public:
338
339       oneOf:
340       - type: null
341       - type: string
342       enum:
343         - yes
344         - no
345         - unspecified
346
347     #=====================
348
349     mitigates_environmental_risk:
350
351       oneOf:
352       - type: null
353       - type: string
354       enum:
```

```
355        - yes
356        - no
357        - unspecified
358
359     #====================
360
361     safe_work_plan_link:
362
363       description: identifies a safety work plan by a permanent URL to the document
        ↪  (e.g., corporate Safe Procedure or Toronto Water Safe Operating Procedures)
364       oneOf:
365         - type: null
366         - type: array
367           items:  #URL strings
368             type: string
369
370     #=======================================
371     # Require Information Data Submission Group
372     #=======================================
```

33_PM.yml

```
1    ---
2    $schema: http://json-schema.org/draft-07/schema#
3    title: PM
4    $id:
     ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
5    type: object
6
7    properties:
8
9      ID:
10
11       description: is a read-only, unique, and permanent ID, generated by the system,
     ↪  to identify PM.
12       type: string
13       $comment: This ID is useful for referencing, even when its name changes.
14
15     # ==========
16
17     name:
18
19       description: a short name for the PM, given by the creator of the PM.
20       type: string
21
22     # ==========
23
24     operational_unit:
25
26       description: indicates Toronto Water's operational unit, on the organizational
     ↪  hierarchy, such as Waste and Storm Water Pumping (symbol - WASP) or Humber
     ↪  Wastewater Treatment Plant (symbol - THR).
```

```yaml
27        $ref: "./04_org_or_group.yml"
28
29    # ==========
30
31    supervisor_group:
32
33        description: indicates a sub-group of the operational unit, that is led by a
   ↪   supervisor who is accountable for the performance of the specified work.
34        $ref: "./04_org_or_group.yml"
35
36    # ==========
37
38    crew_assignment:
39
40        description: identifies a crew, under the supervisor group, that is always
   ↪   assigned to perform the specified work
41
42        oneOf:
43          - type: null
44          - $ref: "./04_org_or_group.yml"
45
46        comment: This value will be determined by (and must be consistent with) the
   ↪   maintainer_organization value - situated at a lower level of the organization
   ↪   hierarchy
47
48    # ==========
49
50    member_of_PM_set:
51
52        description: indicates that this PM is a member of a set of related PMs. For
   ↪   example, the PMs for raw water pump 1, 2, 3 are all members of a PM set named
   ↪   Raw Water 5-year Disassembly Maintenance.
53
54        oneOf:
55          - type: null
56          - $ref: "#/definitions/PM_set"
57        rule to add []: only applicable to higher-level PM
58
59    # ==========
60
61    Avantis_PM:
62
63        description: indicate the Avantis PM (the legacy WMS) that this Maximo PM
   ↪   (equivalent to an Avantis PM task) was a part of.
64
65        oneOf:
66          - type: null
67          - type: string
68        read-only: TRUE
69
70        comment: This field can be eliminated in the future.
71
72        work_entity_harmonization: WR(x), WO(x), JP(x)
```

```yaml
73
74    # ==========
75
76    processes_covered_by_PM:
77
78      description: a list of all major process systems covered by the work specified
   ↪  in this PM.
79      read-only: TRUE
80      oneOf:
81        - type: null
82        - type: object
83          properties:
84
85            ranking:
86              type: number
87
88            system_naming:
89              type: string
90
91      rule to add []: only applicable to higher-level PM
92
93
94
95    # Work Specification at a High-level
96    #========================================
97
98    role_to_work_on:
99
100     description: indicates role at which the specified must be performed.
101     oneOf:
102       - type: null
103       - $ref: "../A-entity_record_schema/02_role.yml"
104
105   #================
106
107   asset:
108
109     description: indicates the asset that is being maintained.
110     oneOf:
111       - type: null
112       - $ref: "../A-entity_record_schema/01_asset.yml"
113
114   # ==========
115
116   job_plan:
117
118     description: specifies the job plan for the PM, if there is no further
   ↪  specification within the route.
119
120
121   # PM Structure Specification
122   #========================================
123
```

```yaml
124    parent:

125

126      description: indicates the more comprehensive PM, usually a shut-down PM, that
    ↪    this PM is a part of.

127

128      oneOf:
129        - type: null
130        - $ref: "../A-entity_record_schema/33_PM.yml"

131

132      $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.

133

134

135    # ==========

136

137   route:

138

139      description: a sequential list of work, composed of job plans paired with an
    ↪    asset/role.
140      oneOf:
141        - type: null
142        - type: object
143          properties:

144

145            sequence:
146              type: number

147

148            asset:
149              oneOf:
150                - type: null
151                - $ref: "./01_asset.yml"

152

153            role:
154              oneOf:
155                - type: null
156                - $ref: "./02_role.yml"

157

158            job_plan:
159              oneOf:
160                - type: null
161                - $ref: "./32_job_plan.yml"

162

163      $comment: the implementation could be done with Maximo route object.

164

165    #========================================
166    # Resources
167    #----------------------------------------
168    # Note: Travel time and preparation time are not being recorded explicitly on the
    ↪    PM. Instead they could be recorded as contributory work in the job plan route
```

```yaml
169    #=====================================
170
171    estimated_duration:
172
173      description: is the estimated time to complete the activity in the job plan
174      $ref: "./00_common_definitions.yml#/definitions/frequency_interval_definition"
175
176    #================
177
178    parts_or_material_requirements:
179
180      description: identifies the parts or material required to complete a work order
   ↪   generated from the PM.
181      oneOf:
182        - type: null
183        - type: array
184          items:
185            $ref:
   ↪   "./00_common_definitions.yml#/definitions/item_requirement_definition"
186
187    #================
188
189    tool_requirements:
190
191      description: identifies the tools required to complete a work order generated
   ↪   from the PM.
192      oneOf:
193        - type: null
194        - type: array
195          items:
196            $ref:
   ↪   "./00_common_definitions.yml#/definitions/tool_requirements_definition"
197
198
199    #================
200
201    skill_and_trade_requirements:
202
203      description: identifies the trades and qualifications of each trade needed to
   ↪   complete the work.
204      oneOf:
205      - type: null
206      - type: array
207        items:
208          $ref: "./00_common_definitions.yml#/definitions
   ↪   trade_requirement_definition"
209
210    #================
211
212    service_requirements:
213
214      description: identifies (contracted) service needed to complete to complete a
   ↪   work order generated from the PM.
```

```yaml
215      oneOf:
216        - type: null
217        - type: array
218          items:
219            $ref:
↪  "./00_common_definitions.yml#/definitions/service_requirement_definition"
220


221

222    #=======================================
223    # Work Triggering Specification
224    #---------------------------------------
225    # Note: the specification is not complete for data mapping purposes, it is
↪   complete for requirement gather
226    #=======================================
227

228    next_due_date_based_on:
229

230      type: string
231      enum:
232        - work start date
233        - work completion date
234

235    trigger_condition:  # aka work generation condition
236

237      oneOf:
238      - type: null
239      - $ref: "#/definitions/time-based_trigger_specification"
240      - $ref: "#/definitions/meter-based_trigger_specification"
241


242

243    #=======================================
244    # Legislative Designation
245    #=======================================
246

247    compliance_level:
248

249      $ref: "./00_common_definitions.yml#/definitions/compliance_class"
250

251      work_entity_harmonization: WR(x), WO(_), JP(_)
252

253 #########################################################
254 # LOCAL OBJECT DEFINITIONS
255 #########################################################
256

257 definitions:
258

259    time-based_trigger_specification:
260

261      next_due_date:
262        type: number
263

264      frequency_interval:
265        $ref: "./00_common_definitions.yml#/definitions/frequency_interval_definition"
```

```yaml
266
267    #================
268
269  meter-based_trigger_specification:
270
271    next_meter_reading:
272        type: number
273
274    meter_condition:
275      oneOf:
276        - type: null
277        - $ref:
   "./00_common_definitions.yml#/definitions/meter_condition_definition"
278
```

34_FR_WR_WO.yml

```yaml
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/TWmaximoConfig/main/1-Schemas/A-entity_record_sc
4
5  $comment: In this design, the work request doubles as a failure report.
6
7  properties:
8
9    #===============================
10   # RECORD
11   #===============================
12
13   ID:
14   #----------------------------
15     description: is a read-only unique ID, generated by the system, to uniquely
   identify the record.
16     type: string
17
18     implementation:
19       MX_mapping: WONUM
20
21   311_ticket_ID:
22   #----------------------------
23     implementation:
24       MX_mapping: COTTICKETID
25       D&C_only: true
26
27   311_request_number:
28   #----------------------------
29     implementation:
30       MX_mapping:
31       D&C_only: true
32
```

```yaml
33        $comment: |
34          [] to resolve: there is some uncertainty of this should be either
   ↪   COTREQUESTNUMBER or EXTERNALREFID
35
36    record_type:
37     #----------------------------
38      description: indicates whether this work request is being used to track an
   ↪   failure, without being a request for work to address the failure.
39      type: string
40      enum:
41        - failure report
42        - work request
43        - work order
44
45      rule_spec:
46        - name: inference of current_type from status
47          status: []
48
49    status:
50     #-----------------------------
51      description: indicate the status of failure report, work request, and work order
52      type: string
53      enum:
54        - failure reported
55        - request made
56        - request approved
57        - request cancelled
58        - waiting on resource
59        - ready to schedule
60        - scheduled
61        - in progress
62        - completed
63        - closed
64        - WO cancelled
65
66      implementation:
67        MX_mapping: WO.status
68
69    following-up_on:
70     #----------------------------
71      description: identifies the work order which this record is following up on.
72      read-only: true
73      oneOf:
74        - type: null
75        - $ref: "../A-entity_record_schema/35_work_order.yml"
76
77     #================================
78     # THE OBJECT
79     #================================
80
81    role_to_work_on:
82     #----------------------------
83      description: indicates role at which the specified work is to be performed.
```

```yaml
 84      oneOf:
 85        - type: null
 86        - $ref: "../A-entity_record_schema/02_role.yml"
 87        [] should have name and description, like asset_to_work_on
 88
 89      rule_spec:
 90        - name: mutual exclusion of asset_to_work_on and role_to_work_on values
 91          spec_ID: 01JFK49T43T1TF3HBTBTDPMN5Y
 92          form: short
 93          specification: |
 94            one, but only one, of the following properties can have a non-null value:
 95              - role_to_work_on
 96              - asset_to_work_on
 97
 98  asset_to_work_on:
 99  #---------------------------
100      description: indicates asset on which the specified work is to be performed.
101      oneOf:
102        - type: null
103        - $ref: "../A-entity_record_schema/01_asset.yml"
104          properties:
105            ID:
106            #---------
107              implementation:
108                MX_mapping: WO.ASSETNUM
109            name:
110            #---------
111
112      implementation:
113        MX_mapping: WO.ASSETNUM
114        $comment: |
115          multiple fields may need to represented.
116
117  asset_or_role_criticality:
118  #---------------------------
119      description: is the criticality values inherited from the asset or role being
     ↪  worked on.
120      read-only: true
121      oneOf:
122        - type: null
123        - type: object
124          properties:
125
126            rating:
127              type: number
128
129            description:
130              type: string
131
132  issue_found_at_address:
133  #---------------------------
134      description: indicates the service address in or around which a failed asset
     ↪  requiring maintenance is located.
```

```yaml
135
136      oneOf:
137        - type: null
138        - $ref: MaximoServiceAddressObject
139
140    route:
141     #----------------------------
142      description: a sequential list of work, composed of job plans paired with an
    ↪  asset/role.
143      oneOf:
144        - type: null
145        - type: object
146          properties:
147
148            sequence:
149              type: number
150
151            asset:
152              oneOf:
153                - type: null
154                - $ref: "./01_asset.yml"
155
156            role:
157              oneOf:
158                - type: null
159                - $ref: "./02_role.yml"
160
161            job_plan:
162              $ref: "./32_job_plan.yml"
163
164      $comment: the implementation could be done with Maximo route object.
165
166    #==================
167    # FAILURE REPORTING
168    #==================
169
170    description_of_issue:
171     #----------------------------
172      description: is a factual description of the observable aspects of a single
    ↪  issue..
173      type: string
174
175      implementation:
176        WR_only: true
177
178    observed_problems:
179     #----------------------------
180      description: is what's also known as a common symptom code (e.g., making noise,
    ↪  cannot start, not running).
181      oneOf:
182        - type: null
183        - type: array
184          items:
```

```yaml
185              $ref: "./00_common_definitions.yml#/definitions/failure_code"
186
187        implementation:
188          WR_only: true
189
190        rule_spec:
191          - name: Which problem codes to show
192            spec_ID: 01JFH3ERR08WHJ0E4WRK166WRT
193            form: short
194            specification: []
195
196        $comment: multiple values are allowed
197
198    physical_causes:
199    #----------------------------
200      oneOf:
201        - type: null
202        - type: array
203          items:
204            properties:
205              cause_code:
206                description: denotes a physical-based failure condition (e.g., shaft
     ↪  misalignment).
207                $ref: "./00_common_definitions.yml#/definitions/failure_code"
208                $comment: one failure code per request
209                rule_spec:
210                  - name: Which Cause Codes to Show
211                    spec_ID: 01JFH2F04P28B4EB2HNWA68KN9
212                    form: short
213                    specification: |
214                      - type must be "cause"
215                      - [] more tbd
216                action_spec:
217                  - name: Creating a New Failure Code
218                    spec_ID: 01JFH2NE68W0NCSBKKSYPZRA2Z
219                    form: short
220                    specification: User must be able to specify new failure codes,
221              basis_of_selection:
222                description: indicates how the failure code was derived
223                type: string
224                enum:
225                  - actual observation
226                  - educated guess from signs
227      implementation:
228        WR_only: true
229
230    bread_crumb:
231
232    found_asset_offline_due_to_this_failure:
233    #----------------------------
234      description: indicates whether the asset was offline, because of the failure
235      type: boolean
236      implementation:
```

```yaml
237        WR_only: true
238      $comment: if true, failure reporting is required []Rule
239
240    took_asset_offline_due_to_this_failure:
241    #-----------------------------
242      description: indicates whether the asset had to be taken offline, because of the
    ↪  failure
243      type: boolean
244      implementation:
245        WR_only: true
246
247    #=======================
248    # WORK DETAIL
249    #=======================
250
251    work_title:
252    #-----------------------------
253      description: a short text summarizing the work that is being requested or have
    ↪  been approved to be performed.
254      oneOf:
255        - type: null
256        - type: string
257
258      implementation:
259        MX_mapping: WO.description
260
261    work_specification:
262    #-----------------------------
263      description: a sufficiently detailed description of the work being requested for
    ↪  the approver of the work.
264
265      implementation:
266        MX_mapping: WO.DESCRIPTION_LONGDESCRIPTION
267
268    work_priority:
269    #-----------------------------
270      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 organization's goals.
271
272      implementation:
273        MX_mapping: INTERNALPRIORITY
274
275    job_plan:
276    #-----------------------------
277      description: specifies the job plan for the PM, if there is no further
    ↪  specification within the route.
278
279      oneOf:
280        - type: null
281        - type: object
282          $ref: "../A-entity_record_schema/32_job_plan.yml"
283
```

```yaml
284      action_spec:
285        name: Importing Specifications from a Job Plan
286        form: short
287        id: 01JFVCVT6Q5F62WAHEB001J7SX
288        specification: TBD []
289
290    work_type:
291     #---------------------------
292        description: is the classification at the work order level
293        $ref: "../B-entity_class_object_schema/33_work_type.yml"
294
295        rule_spec:
296          name: Failure Reporting Leads to Investigation or Repair
297          form: short
298          id: 01JFVCZ9Y7G5MWP2G2DADB2G8Z
299          specification: if failure is reported, then work type must either be
   ↪  investigative or repair
300
301    discrete_activity_classification:
302     #---------------------------
303        description: is a classification often inherited from the job plan specified on
   ↪  the work order
304        $ref: "../B-entity_class_object_schema/33_work_type.yml"
305        not_on_WR: true
306
307    site:
308     #---------------------------
309        description: indicates Toronto Water's operational unit, on the organizational
   ↪  hierarchy, such as Waste and Storm Water Pumping (symbol - WASP) or Humber
   ↪  Wastewater Treatment Plant (symbol - THR).
310        $ref: "./04_org_or_group.yml"
311
312        implementation:
313          $comment: WO.SITEID
314
315    maintenance_group:
316     #---------------------------
317        description: indicates a sub-group of the operational unit, that is led by a
   ↪  supervisor who is accountable for performing the work.
318        $ref: "./04_org_or_group.yml"
319
320        rule_spec:
321          name: Inherit the maintenance_group value from either the asset or the role
322          spec_ID: 01JFK43CJBC495TB7Y3H3VP172
323          form: very short
324
325    requires_asset_offline:
326     #-------------------
327        description: indicates the work requires the asset to be offline
328        oneOf:
329          - type: null
330          - type: string
331        enum:
```

```yaml
332        - yes
333        - no
334        - unknown
335
336    rule_spec:
337      - name: Default value of requires_asset_offline is null
338        req_spec_ID: 01JFK2J0HWVWKDK4WWK5RZCXWY
339        form: very short
340
341  #================================
342  # RESPONSIBILITIES AND ASSIGNMENTS
343  #================================
344
345  crew_assignment:
346  #----------------------------
347    description: identifies a crew, under the supervisor group, that is always
↪  assigned to perform the specified work
348
349    oneOf:
350      - type: null
351      - $ref: "./04_org_or_group.yml"
352
353    WO_only: true
354
355    comment: This value will be determined by (and must be consistent with) the
↪  maintainer_organization value - situated at a lower level of the organization
↪  hierarchy
356
357  trades_assignment:
358  #----------------------------
359    description: identifies the individual trades-persons who will be performing the
↪  work order.
360    oneOf:
361      - type: null
362      - type: array
363        items:
364          $ref: "./08_person.yml"
365
366  asset_covered_by_warranty_contract:
367  #---------------------------
368    description: indicates that the asset (or the asset in the role) is currently
↪  covered by a warranty contract.
369    type: boolean
370
371    implementation:
372      MX_mapping: WO.WARRANTYEXIST
373
374    todo []: rule - determine the value from the asset's warranty information.
375
376  warranty_expiration_date:
377  #---------------------------
378    description: indicates the date the warranty expires, if the asset is covered by
↪  a warranty contract.
```

```yaml
379        type: number
380
381        implementation:
382          MX_mapping: WO.WARRANTYEXPDATE
383
384    send_work_to_warranty_contractor:
385     #----------------------------
386        description: a true (or yes) value indicates that the specified work should be
     ↪  performed by the warranty contractor.
387        type: boolean
388        todo []: rule - enable this field, only if asset_covered_by_warranty_contract is
     ↪  true
389
390    asset_covered_by_service_contract:
391     #----------------------------
392        description: indicates that the asset (or the asset in the role) is currently
     ↪  covered by a service contract.
393        type: boolean
394
395    send_work_to_service_contractor:
396     #----------------------------
397        description: a true (or yes) value indicates that the specified work should be
     ↪  performed by a the selected service contractor.
398        oneOf:
399          - type: null
400          - $ref: "../A-entity_record_schema/07_service_item_master.yml"
401
402    supports_a_capital_project:
403     #----------------------------
404        description: indicate that the specified work supports the work being done be a
     ↪  capital project consultant or contractor.
405        type: boolean
406
407    supports_the_capital_project:
408     #----------------------------
409        description: indicates the specific capital project (represented as a work
     ↪  order)
410        oneOf:
411          - type: null
412          - $ref: "../A-entity_record_schema/35_work_order.yml"
413
414     #========================
415     # Resources
416     #========================
417
418    estimated_duration:
419     #--------------------
420        description: is the estimated time required in hours to complete the activity in
     ↪  the job plan
421        oneOf:
422          - type: null
423          - type: number
424
```

```yaml
425    part_or_material_requirements:
426     #----------------------------
427       description: identifies the parts or material required to complete a work order
    ↪   generated from the PM.
428       oneOf:
429         - type: null
430         - type: array
431           items:
432             $ref:
    ↪   "./00_common_definitions.yml#/definitions/item_requirement_definition"
433
434    tool_requirements:
435     #--------------------
436       description: identifies the tools required to complete a work order generated
    ↪   from the PM.
437       oneOf:
438         - type: null
439         - type: array
440           items:
441             $ref:
    ↪   "./00_common_definitions.yml#/definitions/tool_requirements_definition"
442
443    service_requirements:
444     #--------------------
445       description: identifies (contracted) service needed to complete to complete a
    ↪   work order generated from the PM.
446       oneOf:
447         - type: null
448         - type: array
449           items:
450             $ref:
    ↪   "./00_common_definitions.yml#/definitions/service_requirement_definition"
451
452    skill_and_trade_requirements:
453     #----------------------------
454       description: identifies the trades and qualifications of each trade needed to
    ↪   complete the work.
455       oneOf:
456         - type: null
457         - type: array
458           items:
459             $ref: "./00_common_definitions.yml#/definitions
    ↪   trade_requirement_definition"
460
461    #=======
462    # DATES
463    #=======
464
465    issue_reported_date:
466     #--------------------
467       description: is the date that the issue or failure was reported.
468       oneOf:
469         - type: null
```

```yaml
470          - type: number
471
472    work_requested_date:
473     #--------------------
474       description: is the date when the work request was submitted.
475       oneOf:
476         - type: null
477         - type: number
478
479    request_approval_date:
480     #--------------------
481       description: is the date that the work request was approved (and when it became
     ↪  a work order).
482       oneOf:
483         - type: null
484         - type: number
485
486    target_start_date:
487     #--------------------
488       description: is the date when the work should begin (according to a certain
     ↪  service standard).
489       oneOf:
490         - type: null
491         - type: number
492
493    target_completion_date:
494     #--------------------
495       description: is the date when the work should be completed (according to a
     ↪  certain service standard).
496       oneOf:
497         - type: null
498         - type: number
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: number
506
507    scheduled_completion_date:
508     #--------------------
509       description: is the date when the work is scheduled (by a scheduler) to be
     ↪  completed.
510       oneOf:
511         - type: null
512         - type: number
513
514    actual_start_date:
515     #--------------------
516       description: is the date when the work actually began.
517       oneOf:
518         - type: null
```

```yaml
519        - type: number
520
521    actual_completion_date:
522     #--------------------
523        description: is the date when the work was actually completed.
524        oneOf:
525          - type: null
526          - type: number
527
528    cancel_date:
529     #--------------------
530        description: is the date when the work was cancelled (and the record became
    ↪  either a cancelled work order or work request).
531        oneOf:
532          - type: null
533          - type: number
534
535    WO_closing_date:
536     #--------------------
537        description: is the date when the work order was closed.
538        oneOf:
539          - type: null
540          - type: number
541
542     #===============================
543     # Failure Information Inherited
544     #===============================
545
546    member_of_PM_set:
547     #----------------
548        description: indicates that this PM is a member of a set of related PMs. For
    ↪  example, the PMs for raw water pump 1, 2, 3 are all members of a PM set named
    ↪  Raw Water 5-year Disassembly Maintenance.
549
550        oneOf:
551          - type: null
552          - $ref: "#/definitions/PM_set"
553
554    parent_work_order:
555     #----------------
556        description: indicates the more comprehensive PM, usually a shut-down PM, that
    ↪  this PM is a part of.
557        oneOf:
558          - type: null
559          - $ref: "../A-entity_record_schema/33_PM.yml"
560
561        WO_only: true
562
563        rule_spec:
564          name: Work Type of Descendant Work Orders
565          form: short
566          id: 01JFVDM89RVDCE7VBVM7FDQHRD
567          specification: In a work order hierarchy, the top-level work order determines
    ↪  the work type of all descendant work orders.
```

```yaml
568
569       $comment: |
570         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.
571
572    summary_of_previous_issue_reports:
573
574      description: presents a summary of previously reported issues and failures
    ↪   related to this work.
575      oneOf:
576        - type: null
577        - type: object
578          name: compiled_issue_report
579          properties:
580            # ------
581            compiled_text_summary:
582              description: the compilation of all text information in a issue report,
    ↪   including problem code, failure code, and description.
583              oneOf:
584                - type: null
585                - type: array
586                  items:
587                    - type: string
588            # ------
589            photographs:
590              description: photographs in the failure report.
591              oneOf:
592                - type: null
593                - type: array
594                  items:
595                    - type: string # photos are converted to a string in JSON
596                    - oneOf:
597                      - contentMediaType: image/png
598                      - contentMediaType: image/jpg
599
600    mitigates_safety_risk_to_staff:
601    #-----------------------------
602      description: indicates the work has impact on workers' safety
603      oneOf:
604      - type: null
605      - type: string
606      enum:
607        - yes
608        - no
609        - unspecified
610
611    mitigates_safety_or_health_risk_to_public:
612    #-----------------------------------------
613      description: indicates that the work has a direct impact on the well-being of
    ↪   the public
```

```yaml
614      oneOf:
615      - type: null
616      - type: string
617      enum:
618        - yes
619        - no
620        - unspecified
621
622    mitigates_environmental_risk:
623      #----------------------------
624      description: indicates that the work has impact on environmental protection
625      oneOf:
626      - type: null
627      - type: string
628      enum:
629        - yes
630        - no
631        - unspecified
632
633      #======================================
634    # Legislative Designation
635      #======================================
636
637    compliance_class:
638      #----------------------------
639      description: indicate that the completion of the specified work would satisfy
    ↪  some compliance requirement of a certain Level.
640      $ref: "./00_common_definitions.yml#/definitions/compliance_class"
641
642    attachments:
643      #----------------------------
644      description: documents or photographs that provide further supplementary
    ↪  information.
645      oneOf:
646        - type: null
647        - $ref: "../A-entity_record_schema/35_work_order.yml"
648
649      implementation:
650        MX_mapping:
651
652
653  ##############################
654  # MINOR OBJECT DEFINITIONS
655  ##############################
656
657
```

36_work_order_documentation.yml

```yaml
1  $schema: http://json-schema.org/draft-07/schema#
2  title: Work Order Documentation
```

```yaml
3   $id:
    ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/1-Schemas/A-entity_record_schema/
4   type: object
5   properties:

6
7     work_order_reference:

8
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
14    # OPERATIONAL STATUS INFORMATION
15    #================================
16    # Note: in a work order containing children work orders, these information only
    ↪  have to be filled in at the parent level.  [] rule

17
18    asset_offline_at_start:

19
20      description: indicates that the asset was offline when the work began
21      type: boolean
22      $comment:

23
24    asset_brought_back_online:

25
26      description: is only applicable if the asset was offline when the work began;
    ↪  this entry indicates that the work, within the scope of this work order, brought
    ↪  the asset back online
27      type: boolean
28      $comment:

29
30
31    # ACTUAL WORK ORDER RESOURCE USAGE
32    #================================

33
34    revised_description_of_actual_work:

35
36      description: is a revised and more accurate description of the actual work
    ↪  performed on the asset.
37      type: string

38
39    actual_start_time:

40
41
42    actual_completion_time:

43
44
45    actual_wrench_time:

46
47      description: is the actual time taken to complete the work order.
48      $ref: "./00_common_definitions.yml#/definitions/frequency_interval_definition"

49
50
```

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

```
96
97      description:
98      oneOf:
99        - type: null
100       - type: array
101         items:
102           $ref: "../A-entity_record_schema/36_work_documentation.yml"
103
104
105    # FAILURE REPORTS AND FOLLOWUP REQUESTS
106    #================================
107    # 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
```

41_meter.yml

```
1  $schema: "http://json-schema.org/draft-07/schema#"
2  title: Meter
3  $id:
   ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/A-entity_record_schema/41_me
4
5  properties:
6
7
```

Folder: B-entity_class_object_schema

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/TWmaximoConfig/main/1-Schemas/B-entity_class_obj
5  type: object
6
7  ############################################################
8  # 1. Properties
9  ############################################################
10
11 properties:
12
13   allOf:
14     - $ref:
   ↪  "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
15     #inherit the defintions and rules from th
   ↪  "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
```

```yaml
16
17    #====================
18    parent class:
19      $ref: B-entity_class_object_schema/01_asset_item_tool_class_object_schema.yml
20
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
   ↪  used as a part of another asset, and would never be given an asset tag. E.g.,
   ↪  bearing.
27      rule_spec:
28        - name: Do not include only used as a part in the asset classification
29          spec_ID: NJ1E1ZbOgg
30          status: TBS
31      $comment: A false or null value materially mean the same thing.
32
33
34
```

02_role_class.yml

```yaml
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_class.yml
5   type: object
6
7   #############################################################
8   # 1. Properties
9   #############################################################
10
11  properties:
12
13    allOf:
14      - $ref:
   ↪  "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
15      #inherit the defintions and rules from th
   ↪  "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
16
17    parent class:
18      $ref: B-entity_class_object_schema/02_role_class.yml
19      sort order: 1-30
20
21    discrete asset role:
22      oneOf: [type: boolean, type: null]
23      read-only: true
24      description: a role that can be occupied by a single discrete asset.
25
```

```
26    functional structure role:
27      oneOf: [type: boolean, type: null]
28      read-only: true
29      description: a role that cannot be occupied any discrete asset, but can be the
   ↪   parent to other roles.
30
31    defined set of roles:
32      oneOf: [type: boolean, type: null]
33      read-only: true
34      description: a role that can be occupied by a collection of descrete assets.
35
36    compatible asset occupant class(s):
37      oneOf:
38        - type: array
39          items:
40            $ref: "./01_asset.yml"
41        - type: null
42      $comment: |
43        []RULE NJQ6BwsVee: A asset must be an instance of one of classes listed in
   ↪   this field to be allowed to occupy a role under this class.
44
45  ###############################################################
46  # 2. High level rules
47  ###############################################################
48
49  rule_spec:
50    - name: Is an asset role, functional structure role, or defined set of roles
   ↪
51      spec_ID: NyD4XGbuex
52      specification: |
53        if roleClassX is a decendent of "Discrete Asset Role" in the role
   ↪   classification hierarchy:
54          set roleClassX.properties."a discrete asset role" to TRUE
55          set roleClassX.properties."a functional structure role" to FALSE
56          set roleClassX.properties."a defined ser of role" to FALSE
57        elif roleClassX is a decendent of "functional structure role" in the role
   ↪   classification hierarchy:
58          set roleClassX.properties."a discrete asset role" to FALSE
59          set roleClassX.properties."a functional structure role" to TRUE
60          set roleClassX.properties."a defined ser of role" to FALSE
61        elif roleClassX is a decendent of "Defined Set of Roles" in the role
   ↪   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
```

```yaml
3  $schema: "http://json-schema.org/draft-07/schema#"
4  type: object
5
6  properties:
7
8    allOf:
9      - $ref:
   ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
10     #inherit the defintions and rules from th
   ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
11
12   parent class:
13     $ref: spaceClassObject
```

04_org_class.yml

```yaml
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:
   ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
11   #inherit the definitions and rules from th
   ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
12
13   parent class:
14     $ref: orgGroupClassObject
15
```

08_trade_type.yml

```yaml
1
```

101_common_class_definitions.yml

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

```yaml
6
7  # This set of properties are used in all classification objects (e.g., asset,
   ↪  organization, etc.)
8
9  properties:
10
11   #====================
12
13   class name:
14     type: string
15     description: a noun or short noun-phrase name of the class
16
17   #====================
18
19   class name with synonym(s):
20     oneOf:
21       - type: null
22       - type: array
23         items:
24           type: string
25     description: other synonymous names that may be used by a user in search
26
27   #====================
28
29   class definition:
30     oneOf:
31       - type: null
32       - type: string
33
34   #====================
35
36   appliable to individual:
37     type: boolean
38     $comment: |
39       a "FALSE" value indicates that the class is meant to be a structural part of
   ↪  the classifcation tree, and cannot be used to classify any entity (i.e. asset,
   ↪  role, space, etc).
40   #====================
41
42   retired:
43     type: boolean
44     rule_spec:
45       - name: rules on retire classes
46         spec_ID: Vy3qYEZ_ex
47         type: [UI]
48         specification: |
49           a retired class should not appear in the list of classes for user
   ↪  selection. Also, the value of its properties."appliable to individual" must be
   ↪  FALSE.
50         status: draft
```

32_discrete_activity_class.yml

```yaml
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:
   ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
11       #inherit the defintions and rules from th
   ↪ "/1-Schemas/B-entity_class_object_schema/00_common_class_definitions.yml"
12
13   parent class:
14     $ref: discreteActivityClassObject
```

33_work_type.yml

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

Folder: 2-Classification_Trees

01_asset_classification.md

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

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 asterik
    ↪  symbol shall have the appliable 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 asterik
    ↪  symbol shall have the appliable 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 asterik symbol shall have the appliable
    ↪  to individual property, found in the class object, set to false"
```

31_work_type.md

```
1   ## Work Type [^2]
2
3   * corrective / repair
4   * emergency [^1]
5   * investigative
6   * preventive
7   * asset data
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     rule_spec:
50     - name: Valid Assignment of an Asset
51       spec_ID: 01JDCNEFAED17CWF2K851ZAJKW
52       type: [assertion]
53       description: |
54         classes names specified with an asterisk symbol shall have the their
     ↪   .property.appliable_to_individual value set to false
55   ```
56
57
58   ### Footnotes
59   [^1]: more will be added before the final implementation.
60   [^2]: the physical modification of a building or a structure may result in the
     ↪   creation and removal of a space, hence it is singled out.
61   [^3]: the process by which a asset's ownership is transferred from a capital project
     ↪   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
    ↪  entity represents a set of things. In the case of the example - the set of
    ↪  drinkin water treatment plants in TW.
19
20  ## Notes
21
22    [^1]: i.e. {[FCL]}, {[FIS]}, {[FHO]}, {[FHA]}
23    [^2]: the set of 18 pumping stations
24    [^3]: the set of all reservoirs and elevated tanks
25    [^4]: pumping station for waste and storm water
26    [^5]: the set of all wet-whether storage, inline an offline.
27    [^6]: i.e. {[THC]}, {[THR]}, {[TAB]}, {[TNT]}
```

03_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,
    ↪  the entity represents a set of things. In the case of the example - the set
    ↪  spaces within the drinking water treatment plants in TW.
19
20  ## 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

README.md

```
1   ## Note
2
3   The Excel file in this folder contains the attribute/property defintions, including
    ↳   data type and value list specs, for a list of asset_classes commonly encountered
    ↳   or bear legislative significance.
4   To view the file, download it to your computer, using the marked option in the image
    ↳   below.
5
6
7   <kbd>
8     <img
    ↳   src="https://github.com/user-attachments/assets/e62ad9c8-c1fa-401d-8c89-bd4e807372d1">
9   </kbd>
```

01_pump.yml

```
1   ---
2   $schema: "http://json-schema.org/draft-07/schema#"
3   title: asset
4   $id:
    ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5   type: object
6
7   $comment: >
8
9   properties:
10
11    orientation:
12
13      oneOf:
14        - type: null
15        - type: string
16      description: Indicates how the pump is oriented in 3D space
17      enum:
18        - Horizontal
19        - Vertical
20        - Angled
21        - Inverted
22
23      $comment: |
24
25    #=======================
26
27    variable_speed:
28
29      oneOf:
30        - type: null
31        - type: boolean
32      description: Indicates the if the pump has varialbe speed control
33
34      $comment: |
35
36    #=======================
37
38    pump_type:
39
40      oneOf:
41        - type: null
42        - type: string
43      description: Indicates the pump type
44      enum:
45        - "Dynamic, Centrifugal"
46        - "Dynamic, Axial"
47        - "Reciprocating, Piston or Plunger"
48        - "Reciprocating, Diaphram"
49        - "Rotary, Vane"
```

```yaml
50          - "Rotary, Piston"
51          - "Screw Pump"
52          - "Gear Pump"
53
54      $comment: |
55
56    #========================
57
58    max_rpm:
59
60      oneOf:
61        - type: null
62        - type: number
63      description: Indicates the maximum RPM for the pump
64
65      $comment: |
66
67    #========================
68
69    max_flow:
70
71      oneOf:
72        - type: null
73        - type: number
74      description: Indicates the maximum flow rate at the maximum RPM for the pump in
   ↪  L/s
75
76      $comment: |
77
78    #========================
79
80
81    pump_head:
82
83      oneOf:
84        - type: null
85        - type: number
86      description: Indicates the pressure head for the pump in metres
87
88      $comment: |
89
90    #========================
91
92    submersible:
93
94      oneOf:
95        - type: null
96        - type: boolean
97      description: Indicates if the pump is submersible
98
99      $comment: |
100
101    #========================
```

```yaml
102
103
104   drive_coupling:
105
106     oneOf:
107       - type: null
108       - type: string
109     description: Indicates how the drive and pump are coupled together
110     enum:
111       - "Direct Drive"
112       - "Belt Drive"
113       - "Gear Drive"
114       - "Flexible"
115       - "Chain Drive"
116       - "Hydraulic"
117
118     $comment: |
119
120   #=======================
121
122   drive:
123
124     oneOf:
125       - type: null
126       - type: string
127     description: Indicates what the mechanicly drives the pump
128     enum:
129       - "Electric Motor"
130       - "Engine"
131
132     $comment: |
133
134   #=======================
135
136   sealed_bearings:
137
138     oneOf:
139       - type: null
140       - type: boolean
141     description: Indicates if the pump has sealed bearings
142
143     $comment: |
144
145   #=======================
```

02_motor.yml

```yaml
1   ---
2   $schema: "http://json-schema.org/draft-07/schema#"
3   title: asset
4   $id:
      ↪   https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
```

```yaml
type: object

$comment: >

properties:

  type:

    oneOf:
      - type: null
      - type: string
    description: Indicates the motor type
    enum:
      - "AC"
      - "AC, Squirrel Cage Induction"
      - "AC, Wound Rotor Induction"
      - "AC, Synchronous"
      - "DC"
      - "DC, Separately Excited"
      - "DC, Self Excited"
      - "DC, Permanent Magnet"


    $comment: |

  #========================


  voltage:

    oneOf:
      - type: null
      - type: number
    description: Indicates the voltage of the motor in Volts

    $comment: |

  #========================

  horse_power:

    oneOf:
      - type: null
      - type: number
    description: Indicates the horse power of the motor

    $comment: |

  #========================

  nema_frame:

    oneOf:
```

```yaml
58        - type: null
59        - type: string
60      description: Indicates the NEMA frame type for the motor
61      enum:
62        - "42"
63        - "48"
64        - "56"
65        - "66"
66        - "182"
67        - "184"
68        - "213"
69        - "215"
70        - "1412AT"
71        - "143T"
72        - "145T"
73        - "146AT"
74        - "148AT"
75        - "149AT"
76        - "182AT"
77        - "182T"
78        - "184T"
79        - "186ACY"
80        - "186AT"
81        - "189AT"
82        - "203#"
83        - "204#"
84        - "2110AT"
85        - "213T"
86        - "215T"
87        - "219AT"
88        - "224#"
89        - "225#"
90        - "254#"
91        - "254T"
92        - "254U"
93        - "256T"
94        - "256U"
95        - "284#"
96        - "284T"
97        - "284TS"
98        - "284U"
99        - "286T"
100       - "286TS"
101       - "286U"
102       - "324#"
103       - "324T"
104       - "324TS"
105       - "324U"
106       - "326#"
107       - "326T"
108       - "326TS"
109       - "326U"
110       - "364#"
```

```yaml
111        - "364S#"
112      - "364T"
113        - "364TS"
114        - "364U"
115        - "365#"
116        - "365T"
117        - "365TS"
118        - "365U"
119        - "404T"
120        - "404TS"
121        - "404U"
122        - "405T"
123        - "405TS"
124        - "405U"
125        - "444T"
126        - "444TS"
127        - "444U"
128        - "445T"
129        - "445TS"
130        - "445U"
131        - "447T&&"
132        - "447TS&&"
133        - "449T"
134        - "449TS"
135        - "48H"
136        - "56H"
137        - "56HZ"
138        - "L182ACY"
139        - "L186AT"
140
141
142      $comment: |
143
144    #=========================
145
146
147    nema_enclosure:
148
149      oneOf:
150        - type: null
151        - type: string
152      description: Indicates what the NEMA enclosure type for the motor
153      enum:
154        - ODP
155        - TEFC
156        - TENV
157        - TEAO
158        - TEWD
159        - EXPL
160        - HAZ
161
162
163      $comment: |
```

```
164
165     #========================
166
167     sealed_bearings:
168
169       oneOf:
170         - type: null
171         - type: boolean
172       description: Indicates if the motor has sealed bearings
173
174       $comment: |
175
176     #========================
```

03_valve.yml

```
1   ---
2   $schema: "http://json-schema.org/draft-07/schema#"
3   title: asset
4   $id:
    ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5   type: object
6
7   $comment: >
8
9   properties:
10
11    type:
12
13      oneOf:
14        - type: null
15        - type: string
16      description: Indicates the valve type
17      enum:
18        - "Ball"
19        - "Butterfly"
20        - "Cone"
21        - "Diaphragm"
22        - "Gate Valve"
23        - "Globe Valve"
24        - "Knife Valve"
25        - "Needle Valve"
26        - "Pinch Valve"
27        - "Plug Valve"
28
29      $comment: may be redundent to lablel them valve in the valve type
30
31    #========================
32
33    special_type:
34
```

```
35        oneOf:
36          - type: null
37          - type: string
38        description: Indicates what type of specialised valve it is if it is a
   ↪    specialised valve
39        enum:
40          - "Air release valve"
41          - "Backflow Preventer"
42          - "Check"
43          - "Pressure Relief Valve"
44          - "Solenoid"
45
46
47        $comment: may be redundent to lablel them valve in the valve type
48
49      #========================
50
51      size:
52
53        oneOf:
54          - type: null
55          - type: number
56        description: Indicates the valve size in inches
57
58        $comment: |
59
60      #========================
61
62      ansi_type:
63
64        oneOf:
65          - type: null
66          - type: string
67        description: Indicates the ansi type for the valve
68        enum:
69          - 150
70          - 300
71          - 400
72          - 600
73          - 900
74          - 1500
75          - 2500
76          - 4500
77
78        $comment:
79
80      #========================
81
82      ansi_class:
83
84        oneOf:
85          - type: null
86          - type: string
```

```yaml
87        description: Indicates the ansi class for the valve
88        enum:
89          - "A - Standard"
90          - "B - Special"
91          - "Limited"
92
93        $comment:
94
95    #========================
96
97    horse_power:
98
99        oneOf:
100         - type: null
101         - type: number
102       description: Indicates the horse power of the motor
103
104       $comment: |
105
106    #========================
107
108    nema_frame:
109
110       oneOf:
111         - type: null
112         - type: string
113       description: Indicates the NEMA frame type for the motor
114       enum:
115         - "42"
116         - "48"
117         - "56"
118         - "66"
119         - "182"
120         - "184"
121         - "213"
122         - "215"
123         - "1412AT"
124         - "143T"
125         - "145T"
126         - "146AT"
127         - "148AT"
128         - "149AT"
129         - "182AT"
130         - "182T"
131         - "184T"
132         - "186ACY"
133         - "186AT"
134         - "189AT"
135         - "203#"
136         - "204#"
137         - "2110AT"
138         - "213T"
139         - "215T"
```

```
140        -  "219AT"
141        -  "224#"
142        -  "225#"
143        -  "254#"
144        -  "254T"
145        -  "254U"
146        -  "256T"
147        -  "256U"
148        -  "284#"
149        -  "284T"
150        -  "284TS"
151        -  "284U"
152        -  "286T"
153        -  "286TS"
154        -  "286U"
155        -  "324#"
156        -  "324T"
157        -  "324TS"
158        -  "324U"
159        -  "326#"
160        -  "326T"
161        -  "326TS"
162        -  "326U"
163        -  "364#"
164        -  "364S#"
165        -  "364T"
166        -  "364TS"
167        -  "364U"
168        -  "365#"
169        -  "365T"
170        -  "365TS"
171        -  "365U"
172        -  "404T"
173        -  "404TS"
174        -  "404U"
175        -  "405T"
176        -  "405TS"
177        -  "405U"
178        -  "444T"
179        -  "444TS"
180        -  "444U"
181        -  "445T"
182        -  "445TS"
183        -  "445U"
184        -  "447T&&"
185        -  "447TS&&"
186        -  "449T"
187        -  "449TS"
188        -  "48H"
189        -  "56H"
190        -  "56HZ"
191        -  "L182ACY"
192        -  "L186AT"
```

```yaml
193
194
195      $comment: |
196
197    #=========================
198
199
200    nema_enclosure:
201
202      oneOf:
203        - type: null
204        - type: string
205      description: Indicates what the NEMA enclosure type for the motor
206      enum:
207        - ODP
208        - TEFC
209        - TENV
210        - TEAO
211        - TEWD
212        - EXPL
213        - HAZ
214
215
216      $comment: |
217
218    #=========================
219
220    sealed_bearings:
221
222      oneOf:
223        - type: null
224        - type: boolean
225      description: Indicates if the motor has sealed bearings
226
227      $comment: |
228
229    #=========================
230
231
232    cwp:
233
234      oneOf:
235        - type: null
236        - type: number
237      description: Indicates the cold working pressure of the valve in psi
238
239      $comment: |
240
241    #=========================
242
243
244    nominal_pressure:
245
```

```yaml
246      oneOf:
247        - type: null
248        - type: number
249      description: Indicates the nominal pressure of the valve in psi
250
251      $comment: |
252
253    #========================
254
255
256    valve_body:
257
258      oneOf:
259        - type: null
260        - type: string
261      description: Indicates the valve body material
262      enum:
263        - "Carbon Steel"
264        - "Stainless Steel"
265        - "Duplex"
266        - "Alloy"
267        - "Composite"
268        - "Titanium"
269
270      $comment: |
271
272    #========================
273
274    actuator_type:
275
276      oneOf:
277        - type: null
278        - type: string
279      description: Indicates the type of actuator
280      enum:
281        - Electric
282        - Pneumatic
283        - Hydraulic
284        - Manual
285
286      $comment: |
287
288    #========================
289
290    stem_seal:
291
292      oneOf:
293        - type: null
294        - type: string
295      description: Indicates the type of stem seal for the valve
296      enum:
297        - "Duplex"
298        - "Lip Seal"
```

```
299       - "O-ring"
300       - "Stuffing Box"
301
302    $comment: |
303
304  #========================
305
306
```

04_breaker.yml

```
1   ---
2   $schema: "http://json-schema.org/draft-07/schema#"
3   title: asset
4   $id:
    ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5   type: object
6
7   $comment: >
8
9   properties:
10
11    type:
12
13      oneOf:
14        - type: null
15        - type: string
16      description: Indicates the breaker type
17      enum:
18        - "Insulated Case"
19        - "Metal Clad or Enclosed"
20        - "Molded Case"
21
22      $comment:
23
24    #========================
25
26    max_voltage:
27
28      oneOf:
29        - type: null
30        - type: number
31      description: Indicates what the maximum continuous voltage rating for the
    ↪  breaker in Volts
32
33      $comment:
34
35    #========================
36
37    max_amperage:
38
```

```yaml
39        oneOf:
40          - type: null
41          - type: number
42        description: Indicates what the maximum continuous current rating for the
   ↪  breaker in Amps
43
44        $comment: |
45
46      #=======================
47
48    main_contactor:
49
50        oneOf:
51          - type: null
52          - type: string
53        description: Indicates the ansi type for the valve
54        enum:
55          - "Air Insulated"
56          - "Air Insulated, Air Blast"
57          - "Vacuum Insulated"
58          - "Oil Insultated"
59          - "Gas Insultated"
60
61        $comment:
62
63      #=======================
64
```

05_starter.yml

```yaml
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: asset
4  $id:
   ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5  type: object
6
7  $comment: >
8
9  properties:
10
11    voltage_rating:
12
13      oneOf:
14        - type: null
15        - type: number
16      description: Indicates the continuous voltage rating for the starter in Volts
17
18      $comment:
19
20    #=======================
```

```yaml
21
22    current_rating:
23
24      oneOf:
25        - type: null
26        - type: number
27      description: Indicates the continuous current rating for the starter in Amps
28
29      $comment:
30
31    #=======================
32
33    vfd_present:
34
35      oneOf:
36        - type: null
37        - type: boolean
38      description: Indicates the presence of a variable frequency drive within the
      starter
39
40      $comment:
41
42    #=======================
43
44    soft_start_present:
45
46      oneOf:
47        - type: null
48        - type: boolean
49      description: Indicates the function of a soft starter in  the starter
50
51      $comment:
52
53    #=======================
54
55    main_contactor:
56
57      oneOf:
58        - type: null
59        - type: string
60      description: Indicates the ansi type for the valve
61      enum:
62        - "Air Insulated"
63        - "Air Insulated, Air Blast"
64        - "Vacuum Insulated"
65        - "Oil Insultated"
66        - "Gas Insultated"
67
68      $comment:
69
70    #=======================
71
```

06_transformer.yml

```yaml
1   ---
2   $schema: "http://json-schema.org/draft-07/schema#"
3   title: asset
4   $id:
    ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5   type: object
6
7   $comment: >
8
9   properties:
10
11    voltage_primary:
12
13      oneOf:
14        - type: null
15        - type: number
16      description: Indicates the primary voltage rating for the transformer in Volts
17
18      $comment:
19
20    #========================
21
22    voltage_secondary:
23
24      oneOf:
25        - type: null
26        - type: number
27      description: Indicates the secondary voltage rating for the transformer in Volts
28
29      $comment:
30
31    #========================
32
33    power_rating:
34
35      oneOf:
36        - type: null
37        - type: number
38      description: Indicates the power rating for the transformer in kVA
39
40      $comment:
41
42    #========================
43
44    oil_filled:
45
46      oneOf:
47        - type: null
48        - type: boolean
49      description: Indicates the requirement for oil cooling for the transformer
```

```
50
51        $comment:
52
53     #========================
54
55     pressure_relay:
56
57        oneOf:
58          - type: null
59          - type: boolean
60        description: Indicates the presence of a sudden pressure relay
61
62        $comment:
63
64     #========================
65
66     cooling_air_fan:
67
68        oneOf:
69          - type: null
70          - type: boolean
71        description: Indicates the presence of a cooling air fan(s)
72
73        $comment:
74
75     #========================
76
77     coolant_pump:
78
79        oneOf:
80          - type: null
81          - type: boolean
82        description: Indicates the presence of a coolant pump
83
84        $comment:
85
86     #========================
87
88     gas_monitor:
89
90        oneOf:
91          - type: null
92          - type: boolean
93        description: Indicates the presence of a gas monitor
94
95        $comment:
96
97     #========================
98
```

07_hvac.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: asset
4  $id:
   ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5  type: object
6
7  $comment: >
8
9  properties:
10
11   cfm_rating:
12
13     oneOf:
14       - type: null
15       - type: number
16     description: Indicates the capacity rating for the hvac unit in CFM
17
18     $comment:
19
20   #========================
21
22   static_pressure:
23
24     oneOf:
25       - type: null
26       - type: number
27     description: Indicates the static pressure for the hvac unit in inch water
   ↪  column
28
29     $comment:
30
31   #========================
32
33   heating:
34
35     oneOf:
36       - type: null
37       - type: boolean
38     description: Indicates the presence of a heating function in the hvac unit such
   ↪  as heating coil or gasburner
39
40     $comment:
41
42   #========================
43
44   cooling:
45
46     oneOf:
47       - type: null
```

```yaml
48       - type: boolean
49     description: Indicates the presence of a cooling function in the hvac unit such
   ↪  as cooling coil
50
51     $comment:
52
53   #========================
54
55   dehumidifier:
56
57     oneOf:
58       - type: null
59       - type: boolean
60     description: Indicates the presence of a dehumidifier
61
62     $comment:
63
64   #========================
65
66   maintainable_damper:
67
68     oneOf:
69       - type: null
70       - type: boolean
71     description: Indicates the presence of a maintainable damper
72     $comment:
73
74   #========================
75
76   belts_present:
77
78     oneOf:
79       - type: null
80       - type: boolean
81     description: Indicates the presence of replacable belts
82
83     $comment:
84
85   #========================
86
```

08_blower_fan.yml

```yaml
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: asset
4  $id:
   ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5  type: object
6
7  $comment: >
```

```yaml
properties:

  cfm_rating:

    oneOf:
      - type: null
      - type: number
    description: Indicates the capacity rating for the hvac unit in CFM

    $comment:

  #========================

  static_pressure:

    oneOf:
      - type: null
      - type: number
    description: Indicates the static pressure for the hvac unit in inch water
      column

    $comment:

  #========================

  drive_coupling:

    oneOf:
      - type: null
      - type: string
    description: Indicates the type of drive coupling
    enum:
      - "Direct Drive"
      - "Belt Drive"
      - "Gear Drive"
      - "Flexible"
      - "Chain Drive"
      - "Hydraulic"

    $comment:

  #========================

  sealed_bearings:

    oneOf:
      - type: null
      - type: boolean
    description: Indicates the presence of sealed of shielded bearings

    $comment:
```

```
60    #========================
```

09_compressor.yml

```
1   ---
2   $schema: "http://json-schema.org/draft-07/schema#"
3   title: asset
4   $id:
    ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5   type: object
6
7   $comment: >
8
9   properties:
10
11    rated_pressure:
12
13      oneOf:
14        - type: null
15        - type: number
16      description: Indicates the capacity rating for the compressor in kPa
17
18      $comment:
19
20    #========================
21
22    rated_flow:
23
24      oneOf:
25        - type: null
26        - type: number
27      description: Indicates the capacity rating for the compressor in SCMH
28
29      $comment:
30
31    #========================
32
33    drive_coupling:
34
35      oneOf:
36        - type: null
37        - type: string
38      description: Indicates the type of drive coupling
39      enum:
40        - "Direct Drive"
41        - "Belt Drive"
42        - "Gear Drive"
43        - "Flexible"
44        - "Chain Drive"
45        - "Hydraulic"
46
```

```
47      $comment:
48
49    #=======================
50
51    sealed_bearings:
52
53      oneOf:
54        - type: null
55        - type: boolean
56      description: Indicates the presence of sealed of shielded bearings
57
58      $comment:
59
60    #=======================
```

10_generator.yml

```
1    ---
2    $schema: "http://json-schema.org/draft-07/schema#"
3    title: asset
4    $id:
     ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5    type: object
6
7    $comment: >
8
9    properties:
10
11
12    mobile:
13
14      oneOf:
15        - type: null
16        - type: boolean
17      description: Indicates if the generator is mobile
18
19      $comment:
20
21    #=======================
22
23    power_rated:
24
25      oneOf:
26        - type: null
27        - type: number
28      description: Indicates the power rating in kilo watts
29
30      $comment:
31
32    #=======================
33
```

```yaml
34   voltage_rating:
35
36     oneOf:
37       - type: null
38       - type: number
39     description: Indicates the capacity rating for the generator in Volts
40
41     $comment:
42
43   #========================
44
45   drive_type:
46
47     oneOf:
48       - type: null
49       - type: string
50     description: Indicates the type of drive
51     enum:
52       - "Engine, Diesel or Bio-diesel"
53       - "Engine, Natural Gas"
54       - "Turbine"
55
56     $comment:
57
58   #========================
59
60   brushless:
61
62     oneOf:
63       - type: null
64       - type: boolean
65     description: Indicates the presence of brushes within the generator
66
67     $comment:
68
69   #========================
70
71   test_load:
72
73     oneOf:
74       - type: null
75       - type: boolean
76     description: Indicates the presence of an electrical connection to attach a test
   ↪  load
77
78     $comment:
79
80   #========================
81
82   drive_coupling:
83
84     oneOf:
85       - type: null
```

```yaml
86        - type: string
87      description: Indicates the type of drive coupling
88      enum:
89        - "Direct Drive"
90        - "Belt Drive"
91        - "Gear Drive"
92        - "Flexible"
93        - "Chain Drive"
94        - "Hydraulic"
95
96      $comment:
97
98    #========================
99
100   sealed_bearings:
101
102     oneOf:
103       - type: null
104       - type: boolean
105     description: Indicates the presence of sealed of shielded bearings
106
107     $comment:
108
109   #========================
```

11_ups.yml

```yaml
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: asset
4  $id:
   ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5  type: object
6
7  $comment: >
8
9  properties:
10
11
12   fls:
13
14     oneOf:
15       - type: null
16       - type: boolean
17     description: Indicates if the UPS directly supports any part of the fire
   ↪  supression or any life safety systems
18
19     $comment:
20
21   #========================
22
```

```yaml
23    battery_capacity:
24
25      oneOf:
26        - type: null
27        - type: number
28      description: Indicates the power rating in kilo watt hours
29
30      $comment:
31
32    #========================
33
34    voltage_output:
35
36      oneOf:
37        - type: null
38        - type: number
39      description: Indicates the output voltage of the UPS in Volts
40
41      $comment:
42
43    #========================
44
45    battery_type:
46
47      oneOf:
48        - type: null
49        - type: string
50      description: Indicates the type of drive
51      enum:
52        - "Sealed / Valve Regulated Lead Acid"
53        - "Flooded / Vented Lead Acid"
54        - "NiCad"
55        - "Li-ion"
56
57      $comment:
58
59    #========================
60
61    intergrated_charger:
62
63      oneOf:
64        - type: null
65        - type: boolean
66      description: Indicates that the UPS is physically integrated with charger
67
68      $comment:
69
70    #========================
```

12_boiler.yml

```yaml
---
$schema: "http://json-schema.org/draft-07/schema#"
title: asset
$id:
  ↪ https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
type: object

$comment: >

properties:

  energy_source:

    oneOf:
      - type: null
      - type: string
    description: Indicates the fuel source for the boiler
    enum:
      - "Natural Gas"
      - "Biogas / Digester Gas"
      - "Propane"
      - "Electric"
      - "Fuel Oil"

    $comment:

  #=======================

  boiled_medium:

    oneOf:
      - type: null
      - type: string
    description: Indicates the boiled/heated medium within the boiler
    enum:
      - Water
      - Steam
      - Glycol

    $comment:

  #=======================

  heat_rate:

    oneOf:
      - type: null
      - type: number
    description: Indicates the rated heat rate in the boiler in kilo Watts

```

```yaml
50        $comment:
51
52     #=======================
53
54     max_pressure:
55
56        oneOf:
57          - type: null
58          - type: number
59        description: Indicates the maximum operating pressure for the boiler in psi
60
61        $comment:
62
63     #=======================
64
65     max_temperature:
66
67        oneOf:
68          - type: null
69          - type: number
70        description: Indicates the maximum operating temperature for the boiler in
    ↪  degrees celsius
71
72        $comment:
73
74     #=======================
75
76     heated_surface:
77
78        oneOf:
79          - type: null
80          - type: number
81        description: Indicates the heating surface area of the boiler in meters squared
82
83        $comment:
84
85     #=======================
86
87     capacity:
88
89        oneOf:
90          - type: null
91          - type: number
92        description: Indicates the size capacity of the boiler in Litres
93
94        $comment:
95
96     #=======================
97
98     tssa_crn:
99
100       oneOf:
101         - type: null
```

```
102        - type: string
103      description: Indicates the CRN number issued by the TSSA
104
105      $comment:
106
107    #========================
```

13_pressure_vessel.yml

```
1  ---
2  $schema: "http://json-schema.org/draft-07/schema#"
3  title: asset
4  $id:
   ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5  type: object
6
7  $comment: >
8
9  properties:
10
11   contained_medium:
12
13     oneOf:
14       - type: null
15       - type: string
16     description: Indicates the medium within the pressure vessel
17     enum:
18       - "Water"
19       - "Steam"
20       - "Glycol"
21       - "Refrigerant"
22       - "Compressed Air"
23       - "Digester Gas"
24       - "Ozone"
25
26     $comment:
27
28    #========================
29
30   capacity:
31
32     oneOf:
33       - type: null
34       - type: number
35     description: Indicates the size capacity of the pressure vessel in Litres
36
37     $comment:
38
39    #========================
40
41   max_pressure:
```

```
42
43       oneOf:
44         - type: null
45         - type: number
46       description: Indicates the maximum operating pressure for the pressure vessel in
   ↪  psi
47
48       $comment:
49
50    #========================
51
52    tssa_crn:
53
54       oneOf:
55         - type: null
56         - type: string
57       description: Indicates the CRN number issued by the TSSA
58
59       $comment:
60
61    #========================
```

14_pressure_piping.yml

```
1   ---
2   $schema: "http://json-schema.org/draft-07/schema#"
3   title: asset
4   $id:
    ↪  https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specificat
5   type: object
6
7   $comment: >
8
9   properties:
10
11    max_size:
12
13       oneOf:
14         - type: null
15         - type: number
16       description: Indicates the maximum pipe size in inches
17
18       $comment:
19
20    #========================
21
22    max_pressure:
23
24       oneOf:
25         - type: null
26         - type: number
```

```yaml
27        description: Indicates the maximum working pressure within the pressure piping
↪    in kPa
28
29        $comment:
30
31    #========================
32
33    max_temperature:
34
35        oneOf:
36          - type: null
37          - type: number
38        description: Indicates the maximum medium temperature within the pressure piping
↪    in degrees celsius
39
40        $comment:
41
42    #========================
43
44    contained_medium:
45
46        oneOf:
47          - type: null
48          - type: string
49        description: Indicates the medium within the pressure piping
50        enum:
51          - "Water"
52          - "Steam"
53          - "Glycol"
54          - "Refrigerant"
55          - "Compressed Air"
56          - "Digester Gas"
57          - "Ozone"
58
59        $comment:
60
61    #========================
62
63    special_application:
64
65        oneOf:
66          - type: null
67          - type: string
68        description: Indicates the speical application required for pressure piping
69        enum:
70          - "piping in fire protection system"
71          - "piping in heating system"
72          - "piping in refrigeration system"
73          - "compressed air piping"
74          - "hot oil piping"
75          - "buried water piping"
76
77        $comment:
```

```
78
79     #========================
80
81     tssa_crn:
82
83       oneOf:
84         - type: null
85         - type: string
86       description: Indicates the CRN number issued by the TSSA
87
88       $comment:
89
90     #========================
```

15_instrumentation.yml

```
1    ---
2    $schema: "http://json-schema.org/draft-07/schema#"
3    title: asset
4    $id:
     ↪  "https://raw.githubusercontent.com/TW-ASMP/TWmaximoConfig/main/4-Class_Dependent_Specifica
5    type: object
6
7    properties:
8
9      parameter:
10       oneOf:
11         - type: null
12         - type: string
13       description: Indicates the parameter that is being measured
14       enum:
15         - "Density"
16         - "Flow Rate"
17         - "Humidity"
18         - "Level"
19         - "Current"
20         - "Power"
21         - "Position"
22         - "Pressure"
23         - "Speed"
24         - "Temperature"
25         - "Torque"
26         - "UV"
27         - "Vibration"
28         - "Weight"
29         - "Specific Gravity"
30         - "Ammonia"
31         - "Carbon Monoxide"
32         - "Chloramination"
33         - "Chlorine"
34         - "Dissolved Oxygen"
```

```yaml
35            - "Fluoride"
36            - "Methane/LEL"
37            - "Total Hydrocarbon"
38            - "ORP"
39            - "Ozone"
40            - "Particulate"
41            - "PH"
42            - "Sulphite"
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    allOf:
54      - if:
55          properties:
56            parameter:
57              const: "Density"
58        then:
59          properties:
60            uom:
61              enum:
62                - "Kilograms Per Cubic Meter (kg/m³)"
63                - "Grams Per Cubic Centimeter (g/cm³)"
64                - "Grams Per Milliliter (g/mL)"
65                - "Pounds Per Cubic Foot (lb/ft³)"
66
67      - if:
68          properties:
69            parameter:
70              const: "Flow Rate"
71        then:
72          properties:
73            uom:
74              enum:
75                - "Liters Per Second (L/s)"
76                - "Cubic Meters Per Second (m³/s)"
77                - "Gallons Per Minute (GPM)"
78                - "Cubic Feet Per Minute (CFM)"
79                - "Liters Per Hour (L/h)"
80                - "Standard Cubic Feet Per Minute (SCFM)"
81
82      - if:
83          properties:
84            parameter:
85              const: "Humidity"
86        then:
```

```yaml
 87        properties:
 88          uom:
 89            enum:
 90              - "Percentage (%)"
 91              - "Grams Per Milliliter (g/mL)"
 92              - "Milligrams Per Liter (mg/L)"
 93              - "Parts Per Million (ppm)"
 94              - "Parts Per Billion (ppb)"
 95
 96    - if:
 97        properties:
 98          parameter:
 99            const: "Level"
100      then:
101        properties:
102          uom:
103            enum:
104              - "Centimeters (cm)"
105              - "Meters (m)"
106              - "Inch (in)"
107              - "Percentage (%)"
108              - "Feet (ft)"
109
110    - if:
111        properties:
112          parameter:
113            const: "Current"
114      then:
115        properties:
116          uom:
117            enum:
118              - "Ampere (A)"
119              - "Milliampere (mA)"
120
121    - if:
122        properties:
123          parameter:
124            const: "Power"
125      then:
126        properties:
127          uom:
128            enum:
129              - "Watt (W)"
130              - "kilowatt (kW)"
131              - "Megawatt (MW)"
132
133    - if:
134        properties:
135          parameter:
136            const: "Position"
137      then:
138        properties:
139          uom:
```

```
140            enum:
141              - "Centimeters (cm)"
142              - "Meters (m)"
143              - "Inch (in)"
144              - "Millimeter (mm)"
145              - "Feet (ft)"
146              - "Degree (°)"
147              - "Randians (rad)"
148              - "Unitless"
149
150    - if:
151        properties:
152          parameter:
153            const: "Pressure"
154      then:
155        properties:
156          uom:
157            enum:
158              - "Pascal (Pa)"
159              - "Kilopascal (kPa)"
160              - "Bar"
161              - "Atmosphere (atm)"
162              - "Pounds Per Square Inch (PSI)"
163              - "Millimeter of Mercury (mmHg)"
164              - "Millimeter of Water (mmH20)"
165              - "Inch of Water (\"WC)"
166
167    - if:
168        properties:
169          parameter:
170            const: "Speed"
171      then:
172        properties:
173          uom:
174            enum:
175              - "Meters Per Second (m/s)"
176              - "Kilometers Per Hour (km/h)"
177              - "Feet Per Second (ft/s)"
178              - "Mile Per Hour (mph)"
179              - "Revolutions Per Minute (RPM)"
180
181    - if:
182        properties:
183          parameter:
184            const: "Temperature"
185      then:
186        properties:
187          uom:
188            enum:
189              - "Degree Celsius (°C)"
190              - "Degree Fahrenheit (°F)"
191
192
```

```yaml
193    - if:
194        properties:
195          parameter:
196            const: "Torque"
197      then:
198        properties:
199          uom:
200            enum:
201              - "Newton-meters (N·m)"
202              - "Foot-pounds (ft·lb)"
203
204    - if:
205        properties:
206          parameter:
207            const: "UV"
208      then:
209        properties:
210          uom:
211            enum:
212              - "Watts Per Square Meter (W/m²)"
213              - "Percentage (%)"
214
215    - if:
216        properties:
217          parameter:
218            const: "Vibration"
219      then:
220        properties:
221          uom:
222            enum:
223              - "Meters Per Second (m/s)"
224              - "Centimeters Per Second (cm/s)"
225              - "Feet Per Seconf (ft/s)"
226              - "Inch Per Second (in/s)"
227              - "Meters Per Second Square (m/s2)"
228              - "Centimeters Per Second Square (cm/s2)"
229              - "Feet Per Second Square (ft/s2)"
230              - "Inch Per Second Square (in/s2)"
231              - "Hertz (Hz)"
232
233    - if:
234        properties:
235          parameter:
236            const: "Weight"
237      then:
238        properties:
239          uom:
240            enum:
241              - "Grams (g)"
242              - "Kilograms (kg)"
243              - "Pounds (lb)"
244              - "Metric Tons (tonne)"
245
```

```yaml
246      - if:
247          properties:
248            parameter:
249              const: "Specific Gravity"
250        then:
251          properties:
252            uom:
253              enum:
254                - "Unitless"
255
256      - if:
257          properties:
258            parameter:
259              const: "Ammonia"
260        then:
261          properties:
262            uom:
263              enum:
264                - "Parts Per Million (ppm)"
265                - "Parts Per Billion (ppb)"
266                - "Milligrams Per Cubic Meter (mg/m³)"
267                - "Percentage (%)"
268                - "Micrograms Per Cubic Meter (µg/m³)"
269
270      - if:
271          properties:
272            parameter:
273              const: "Carbon Monoxide"
274        then:
275          properties:
276            uom:
277              enum:
278                - "Parts Per Million (ppm)"
279                - "Parts Per Billion (ppb)"
280                - "Milligrams Per Cubic Meter (mg/m³)"
281                - "Percentage (%)"
282                - "Micrograms Per Cubic Meter (µg/m³)"
283
284      - if:
285          properties:
286            parameter:
287              const: "Chloramination"
288        then:
289          properties:
290            uom:
291              enum:
292                - "Grams Per Milliliter (g/mL)"
293                - "Milligrams Per Liter (mg/L)"
294                - "Parts Per Million (ppm)"
295                - "Parts Per Billion (ppb)"
296
297
298      - if:
```

```
299          properties:
300            parameter:
301              const: "Chlorine"
302        then:
303          properties:
304            uom:
305              enum:
306                - "Grams Per Milliliter (g/mL)"
307                - "Milligrams Per Liter (mg/L)"
308                - "Parts Per Million (ppm)"
309                - "Parts Per Billion (ppb)"
310
311      - if:
312          properties:
313            parameter:
314              const: "Dissolved Oxygen"
315        then:
316          properties:
317            uom:
318              enum:
319                - "Grams Per Milliliter (g/mL)"
320                - "Milligrams Per Liter (mg/L)"
321                - "Parts Per Million (ppm)"
322                - "Parts Per Billion (ppb)"
323
324      - if:
325          properties:
326            parameter:
327              const: "Fluride"
328        then:
329          properties:
330            uom:
331              enum:
332                - "Grams Per Milliliter (g/mL)"
333                - "Milligrams Per Liter (mg/L)"
334                - "Parts Per Million (ppm)"
335                - "Parts Per Billion (ppb)"
336
337      - if:
338          properties:
339            parameter:
340              const: "Methane/LEL"
341        then:
342          properties:
343            uom:
344              enum:
345                - "Parts Per Million (ppm)"
346                - "Parts Per Billion (ppb)"
347                - "Milligrams Per Cubic Meter (mg/m³)"
348                - "Percentage (%)"
349                - "Micrograms Per Cubic Meter (µg/m³)"
350
351      - if:
```

```yaml
352        properties:
353          parameter:
354            const: "Total Hydrocarbon"
355      then:
356        properties:
357          uom:
358            enum:
359              - "Grams Per Milliliter (g/mL)"
360              - "Milligrams Per Liter (mg/L)"
361              - "Parts Per Million (ppm)"
362              - "Parts Per Billion (ppb)"
363
364    - if:
365        properties:
366          parameter:
367            const: "ORP"
368      then:
369        properties:
370          uom:
371            enum:
372              - "Grams Per Milliliter (g/mL)"
373              - "Milligrams Per Liter (mg/L)"
374              - "Parts Per Million (ppm)"
375              - "Parts Per Billion (ppb)"
376
377    - if:
378        properties:
379          parameter:
380            const: "Ozone"
381      then:
382        properties:
383          uom:
384            enum:
385              - "Grams Per Milliliter (g/mL)"
386              - "Milligrams Per Liter (mg/L)"
387              - "Parts Per Million (ppm)"
388              - "Parts Per Billion (ppb)"
389
390    - if:
391        properties:
392          parameter:
393            const: "Particulate"
394      then:
395        properties:
396          uom:
397            enum:
398              - "Grams Per Milliliter (g/mL)"
399              - "Milligrams Per Liter (mg/L)"
400              - "Parts Per Million (ppm)"
401              - "Parts Per Billion (ppb)"
402
403    - if:
404        properties:
```

```yaml
405        parameter:
406          const: "PH"
407    then:
408      properties:
409        uom:
410          enum:
411            - "Unitless"
412
413  - if:
414      properties:
415        parameter:
416          const: "Sulphite"
417    then:
418      properties:
419        uom:
420          enum:
421            - "Grams Per Milliliter (g/mL)"
422            - "Milligrams Per Liter (mg/L)"
423            - "Parts Per Million (ppm)"
424            - "Parts Per Billion (ppb)"
425
426  - if:
427      properties:
428        parameter:
429          const: "Sulphur Dioxide"
430    then:
431      properties:
432        uom:
433          enum:
434            - "Parts Per Million (ppm)"
435            - "Parts Per Billion (ppb)"
436            - "Milligrams Per Cubic Meter (mg/m³)"
437            - "Percentage (%)"
438            - "Micrograms Per Cubic Meter (µg/m³)"
439
440  - if:
441      properties:
442        parameter:
443          const: "Suspended Solids"
444    then:
445      properties:
446        uom:
447          enum:
448            - "Grams Per Milliliter (g/mL)"
449            - "Milligrams Per Liter (mg/L)"
450            - "Parts Per Million (ppm)"
451            - "Parts Per Billion (ppb)"
452
453  - if:
454      properties:
455        parameter:
456          const: "Turbity"
457    then:
```

```
458        properties:
459          uom:
460            enum:
461              - "Grams Per Milliliter (g/mL)"
462              - "Milligrams Per Liter (mg/L)"
463              - "Parts Per Million (ppm)"
464              - "Parts Per Billion (ppb)"
```

Folder: 5-Functions