|  |  |  |
| --- | --- | --- |
| WEATHER CLIMATE WATER | **World Meteorological Organization**  **COMMISSION FOR OBSERVATION, INFRASTRUCTURE AND INFORMATION SYSTEMS**  **Third Session** 15 to 19 April 2024, Geneva | **INFCOM-3/INF. 8.3(6a)** |
| Submitted by: Chair of SC-IMT  7.III.2024 |

## EXPERIMENTAL CLIMATE DATA MODEL

### DRAFT WMO CORE CLIMATE DATA MODEL – OBSERVATIONS

# FM 251: WCCDM-OBS

## FM 251–2024 WCCDM-OBS WMO Core Climate Data Model – Observations

### 251–2024.1 Scope and conventions

#### 251–2024.1.1 Scope

WCCDM-OBS may be used to represent and exchange individual observations of the physical environment, such as the observations of the essential climate variables. This model is intended to be used for all exchanges of climate data, from measurements to climate services.

Notes:

(1) The intent of WCCDM-OBS is to allow encoded observations to be packaged in a way that facilitates the exchange of the data via modern WebAPIs, such as the *OGC API Features – Part 1: Core* standard (see <https://ogcapi.ogc.org/features/)>.

(2) The verb ‘may’ has been used to permit the use of the WCCDM-OBS model and its representation on the WMO Information System, this is not intended to replace other requirements such as the use of BUFR or GRIB in the exchange of real time data.

(3) The requirements classes defined herein are intended to be compatible with the *OGC Abstract Specification Topic 20: Observations, measurements and samples*, in particular the logical model defined by the Basic Observations Package Requirements Class (http://www.opengis.net/spec/om3.0/req/obs-basic).

(4) The data makes use of links to external concepts, such as the wmdr:ObservingFacility and wmdr:Equipment, to provide metadata. Figure 251–2024.1 shows the relationship between a WCCDM-OBS object and external data models.

(5) The model is intended to be flexible and extensible. As such, soft typing is used in a number of places and the method of referencing external code lists is specified rather than defining new code lists.

A black background with white text

Description automatically generated

**Figure 251–204.1: Relationship between the WCCDM-OBS and external data models, the names in angular brackets e.g.<<WMDR>>, <<WCMP2>>, <<WCCDM-OBS>> indicate the data model**

#### 251–2021.1.2 Conventions

The normative provisions in this standard are denoted by the URI:

<http://wis.wmo.int/spec/wccdm-obs/1>

All requirements that appear in this document are denoted by partial URIs which are relative to this base URI.

#### 251–2021.1.3 Requirements classes

The requirements classes defined in WCCDM-OBS are listed in Table 251–2024.1.

**Table 251–2024.1. Requirements classes defined in WCCDM-OBS**

|  |  |
| --- | --- |
| Requirements classes | |
| Requirements class | /req/core/json |
| Requirements class | /req/core/vocabularies |
| Requirements class | /req/observation |
| Requirements class | /req/geometry |
| Requirements class | /req/properties |
| Requirements class | /req/result |
| Requirements class | /req/qualityFlag |
| Requirements class | /req/parameter |
| Requirements class | /req/feature |

### 251–2024.2 JSON schema for WCCDM-OBS

251–2024.2.1 A WCCDM-OBS object shall conform with the Entity Relationship Diagram shown in Figure 251–2024.2.

A screenshot of a computer

Description automatically generated

**Figure 251–2024.2: Entity relationship diagram (ERD) of the different objects within the WCCDM-OBS data model. Mandatory elements are indicated by a closed circle.**

Note: hereafter, the different entities from Figure 251–2024.2 are referred to as objects.

251–2024.2.2 A WCCDM-OBS object encoded in JSON shall conform with the requirements class specified in Table 251–2024.2.

Note: the JSON representation is intended for scalar values and is not optimized for coverages, such as remotely sensed data and numerical model output.

**Table 251–2024.2: Requirements class /req/core/json**

|  |  |
| --- | --- |
| Requirements class | |
| /req/core/json | |
| Target type | Data instance |
| Name | JSON schema |
| Requirement | /req/core/json/schema  A WCCDM-OBS object encoded in JSON shall validate against the JSON schema provided at:  http://schemas.wmo.int/wccdm-obs/2024/wccdm-obs.json |

251–2024.2.3 Additional representations of WCCDM-OBS objects may be developed and included in this Manual.

### 251–2024.3 Use of HTTP URIs and vocabulary services

251–2024.3.1 References to categorical values, code table entries and values from controlled vocabularies shall conform with the requirements specified in Table 251–2024.3.

**Table 251–2024.3: Requirements class req/core/vocabularies**

|  |  |
| --- | --- |
| Requirements class | |
| /req/core/vocabularies | |
| Target type | Data instance |
| Name | Vocabularies |
| Requirement | /req/core/vocabularies/uri  References to categorical values, code table entries and values from controlled vocabularies shall be provided via HTTP URIs. |
| Requirement | /req/core/vocabularies/rdf  The HTTP URIs shall resolve to an entry in a controlled vocabulary from a vocabulary service that includes a Resource Description Framework (RDF) representation of the definition. |
| Permission | /per/core/vocabularies/non-rdf  Representations, other than RDF, may also be provided by the vocabulary service. |
| Recommendation | /rec/vocabularies/WMORegistry  The vocabulary service provided by the WMO Codes Registry should be used by default. See note 1. |
| Recommendation | /rec/vocabularies/skosExactMatch  The RDF representation provided by the resolution of the HTTP URI should include essential mapping elements, such as ‘skos:exactMatch’, to enable mapping between different vocabularies. |

Note: The WMO Codes Registry can be accessed via <https://codes.wmo.int>

Example: Use of an HTTP URI and the WMO Codes Registry to report the property observed by the observation.

|  |
| --- |
| "properties": {  ...,  "observedProperty": "http://codes.wmo.int/wmdr/ObservedVariableAtmosphere/224",  ...  } |

### 251–2024.4 Requirements class: Observation

251–2024.4.1 A WCCDM-OBS ‘Observation’ object shall conform to the requirements specified in Table 251–2024.4.

**Table 251–2024.4: Requirements class/req/observation**

|  |  |
| --- | --- |
| Requirements class | |
| /req/observation | |
| Target type | Data instance |
| Name | Observation |
| Dependency | <https://datatracker.ietf.org/doc/html/rfc7946#section-3> |
| Dependency | /req/geometry |
| Dependency | /req/properties |
| Requirement | /req/observation/conformance  An ‘Observation’ object shall include information on the standards it conforms to via the ‘conformsTo’ element. |
| Requirement | /req/observation/conformance/encoding  The ‘conformsTo’ element shall be encoded as an array of HTTP URIs identifying the standards that the object conforms to. |
| Requirement | /req/observation/conformance/wccdm-obs  The ‘conformsTo’ array shall have a minimum size of 1 and shall include the version of the WCCDM-OBS that the object conformsTo. |
| Requirement | /req/observation/geojson  An ‘Observation’ object shall be a valid GeoJSON (RFC 7946) and encoded as defined in this specification. |
| Requirement | /req/observation/identifier  An ‘Observation’ object shall provide an identifier via the ‘id’ element. |
| Requirement | /req/observation/identifier/encoding  The identifier shall be encoded as a string using UTF-8 encoding. |
| Requirement | /req/observation/identifier/uniqueness  The identifier shall be unique within the collection, or dataset, containing the observation. |
| Requirement | /req/observation/type  The element ‘type’ shall have the value "Feature" |
| Requirement | /req/observation/geom/cardinality  An ‘Observation’ object shall provide exactly one ‘Geometry’ object via the ‘geometry’ element. |
| Requirement | /req/observation/properties/cardinality  An ‘Observation’ object shall provide exactly one ‘Properties’ object via the ‘properties’ element. |

Note: this requirements class defines the top level (parent) object for a WCCDM-OBS record.

Example: ‘Observation’ object represented in GeoJSON showing conformance and other main elements. For clarity nested objects have been collapsed and represented by ellipses (…), these objects are defined further in later sections.

|  |
| --- |
| {  "conformsTo": ["http://wis.wmo.int/spec/wccdm-obs/1/req/core/json"],  "id": "1",  "type": "Feature",  "geometry": {...},  "properties": {...}  } |

### 251–2024.5 Requirements class: Geometry

251–2024.5.1 A ‘Geometry’ object shall be a valid GeoJSON Geometry object and conform to the requirements class specified in Table 251.2024.5.

Notes:

(1) The use of GeoJSONs limits the Coordinate Reference System (CRS) to the World Geodetic System 1984 (WGS84).

(2) Permission /per/parameter/additionalProperties allows additional information to be provided with each observation, for example the location of the feature of interest in an alternative CRS or the inclusion of uncertainty estimates for the location.

**Table 251–2024.5: Requirements class /req/geometry**

|  |  |
| --- | --- |
| Requirements class | |
| /req/geometry | |
| Target type | Data instance |
| Name | Geometry |
| Dependency | <https://datatracker.ietf.org/doc/html/rfc7946#section-3.1> |
| Requirement | /req/geometry/featureOfInterest  The geometry provided shall be the location of the feature of interest that is the subject of observation.  See also: /req/feature. |
| Requirement | /req/geometry/type  The geometry type, specified by the ‘type’ element, shall be suitable for the type of observation. |
| Recommendation | /rec/geometry/elevation  Where known, and applicable, the elevation of the feature of interest should be provided as the third element in the ‘coordinates’ array. |

Example: GeoJSON ‘Geometry’ representing a point location (46.22325 N 6.14625 E) with elevation (375 m).

|  |
| --- |
| {  "type": "Point",  "coordinates": [6.14625,46.22325,375]  } |

### 251–2024.6 Requirements class: Properties

251–2024.6.1 A ‘Properties’ object shall conform to the requirements class specified in Table 251–2024.6.

**Table 251–2024.6: Requirements class /req/properties**

|  |  |
| --- | --- |
| Requirements class | |
| /req/properties | |
| Target type | Data instance |
| Name | Properties |
| Dependency | /req/core/vocabularies |
| Dependency | /req/result |
| Dependency | /req/qualityFlag |
| Dependency | /req/parameter |
| Dependency | /req/feature |
| Requirement | /req/properties/result/cardinality  A ‘Properties’ object shall provide exactly one ‘Result’ object via the ‘result’ element. |
| Requirement | /req/properties/resultQuality/cardinality  A ‘Properties’ object shall provide at least one ‘QualityFlag’ object via the ‘resultQuality’ element. The ‘resultQuality’ element shall be encoded as an array. |
| Requirement | /req/properties/parameter/cardinality  A ‘Properties’ object shall provide exactly one ‘Parameter’ object via the ‘parameter’ element. |
| Requirement | /req/properties/feature/cardinality  A ‘Properties’ object shall provide at least one ‘Feature’ object via the ‘featureOfInterest’ element. The ‘featureOfInterest’ shall be encoded as an array. |
| Requirement | /req/properties/host  A ‘Properties’ object shall provide a reference to the station or platform associated with the parent observation via the ‘host’ element. |
| Requirement | /req/properties/host/encoding  The ‘host’ element shall be encoded as an HTTP URI and resolve to a valid wmdr:ObservingFacility object. |
| Recommendation | /rec/properties/host/compilation  The HTTP URI provided by the host element should resolve to a record hosted in one of the global compilations of WIGOS metadata. See note. |
| Recommendation | /rec/properties/observer  When available, a ‘Properties’ object should provide a reference to the sensor or instrument used to make the parent observation via the ‘observer’ element. |
| Requirement | /req/properties/observer/encoding  When the ‘observer’ element is provided it shall be encoded as an HTTP URI and shall resolve to a valid wmdr:Equipment object. |
| Recommendation | /rec/properties/observer/compilation  The HTTP URI provided by the observer element should resolve to a record hosted in one of the global compilations of WIGOS metadata. See note. |
| Requirement | /req/properties/observationType  A ‘Properties’ object shall provide information on the type of observation via the ‘observationType’ element. |
| Requirement | /req/properties/observationType/encoding  The ‘observationType’ shall be encoded as an HTTP URI and shall resolve to an online resource providing a definition of the observation type. |
| Requirement | /req/properties/observedProperty  A ‘Properties’ object shall provide a reference to the observable property that is the subject of observation via the ‘observedProperty’ element. |
| Requirement | /req/properties/observedProperty/encoding  The reference provided by the ‘observedProperty’ element shall be encoded as an HTTP URI and shall resolve to an online resource providing a definition of the observable property. |
| Requirement | /req/properties/observingProcedure  A ‘Properties’ object shall provide information on the procedure used to determine the result of the parent observation via the ‘observingProcedure’ element. |
| Requirement | /req/properties/observingProcedure/encoding  The ‘observingProcedure’ element shall be encoded as an HTTP URI that resolves to an online resource providing that information. |
| Requirement | /req/properties/phenomenonTime  A ‘Properties’ object shall provide the date and time of the parent observation via the ‘phenomenonTime’ element. |
| Requirement | /req/properties/phenomenonTime/encoding  The ‘phenomenonTime’ element shall be encoded as a valid ISO 8601 string and in Universal Coordinated Time (UTC). |
| Requirement | /req/properties/phenomenonTime/type  For instantaneous observations the date and time provided by the ‘phenomenonTime’ element shall be a date time instant.  For temporal averages or durations the date and time provided by the ‘phenomenonTime’ element shall be a date time interval. |
| Requirement | /req/properties/resultTime  A ‘Properties’ object shall provide the time when the result of the parent observation became available or was updated via the ‘resultTime’ element. |
| Requirement | /req/properties/resultTime/encoding  The ‘resultTime’ element shall be a time instant reported in Universal Coordinated Time (UTC) and encoded as a valid ISO 8601 string. |
| Permission | /per/properties/validTime  When the parent observation has an associated period of validity this may be reported via the ‘validTime’ element of the ‘Properties’ object. |
| Requirement | /req/properties/validTime/encoding  When the ‘validTime’ element is provided the period of validity shall be reported as a date time interval in Universal Coordinated Time (UTC) and encoded as a valid ISO 8601 string. |

Note: further information on the global compilations of WIGOS metadata can be found in the Manual on the WMO Integrated Global Observing System (WMO-No. 1160).

Example: ‘Properties’ object showing example values for the different elements. Note that in many cases HTTP URIs are not interpretable without resolving the URI (e.g. see the example from provision 251–2024.3 above). For readability, example.com and identifiers that use plain text have been used rather than published HTTP URIs.

|  |
| --- |
| {  "featureOfInterest": [...],  "host": "http://data.example.com/stations/1",  "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM\_Measurement",  "observedProperty": "http://vocab.example.com/observedProperty/airTemperature",  "observer": "http://data.example.com/equipment/1",  "observingProcedure":  "http://vocab.example.com/observingProcedure/automatic",  "parameter": {...},  "phenomenonTime": "19931–01T00:00Z/19931–02T00:00Z",  "result": {...},  "resultQuality": [...],  "resultTime": "19931–02T05:00Z",  "validTime": null  } |

### 251–2024.7 Requirements class: Result

251–2024.7.1 A ‘Result’ object shall conform to the requirements class specified in Table 251–2024.7.

**Table 251–2024.7: Requirements class /req/result**

|  |  |
| --- | --- |
| Requirements class | |
| /req/result | |
| Target type | Data instance |
| Name | Result |
| Dependency | /req/core/vocabularies |
| Requirement | /req/result/value  A ‘Result’ object shall provide the result value via the ‘value’ element. |
| Requirement | /req/result/value/encoding  The encoding of the ‘value’ element shall be appropriate for the type of observation and as specified by the ‘observationType’ from the parent ’Properties’ object. |
| Requirement | /req/result/units  A ‘Result’ object shall provide the units that the result is reported in via the ‘units’ element. |
| Requirement | /req/result/units/encoding  The ‘units’ element shall be encoded as an HTTP URI and resolve to a definition of the units within a controlled vocabulary or code list. |
| Recommendation | /rec/result/standardUncertainty  When available, a ‘Result’ object should provide an estimate of the standard uncertainty via the ‘standardUncertainty’ element. |
| Requirement | /rec/result/standardUncertainty/units  When the ‘standardUncertainty’ element is provided it shall be provided in the same units as the ‘value’ element and as reported in the ‘units’ element. |

Example: ‘Result’ object showing an observed value of 13.9 degrees Celsius and with a standard uncertainty of 0.2 degrees Celsius. The result represents a point measurement as per the cell methods attribute.

|  |
| --- |
| {  "value": 13.9,  "units": "https://qudt.org/vocab/unit/DEG\_C",  "standardUncertainty": 0.2,  } |

### 251–2024.8 Requirements class: qualityFlag

251–2024.8.1 A ‘QualityFlag’ object shall conform with the requirements class specified in Table 251–2024.8.

**Table 251–2024.8: Requirements class /req/qualityFlag**

|  |  |
| --- | --- |
| Requirements class | |
| /req/qualityFlag | |
| Target type | Data instance |
| Name | Quality flag |
| Dependency | /req/core/vocabularies |
| Requirement | /req/qualityFlag/inScheme  A ‘QualityFlag’ object shall include information on the quality flagging scheme used via the 'inScheme' element. |
| Requirement | /req/qualityFlag/inScheme/encoding  The 'inScheme' element shall be provided as an HTTP URI that resolves to further information on the quality flagging scheme. |
| Requirement | /req/qualityFlag/flag  A 'QualityFlag' object shall include the flag, or test applied, via the 'flag' element. |
| Requirement | /req/qualityFlag/flag/encoding  The 'flag' element shall be encoded as an HTTP URI that resolves to an online resource providing further information on the test applied. |
| Requirement | /req/qualityFlag/flagValue  The result of the quality control applied shall be provided by the 'flagValue' element. |
| Requirement | /req/qualityFlag/flagValue/encoding  The 'flag' element shall be encoded as an HTTP URI that resolves to an online resource providing a definition of the flag. |
| Permission | /per/qualityFlag/additionalProperties  Additional properties may be provided by including additional key / value pairs. |

Note: the ‘QualityFlag’ object is weakly typed due to the wide variety of quality flagging schemes in use across the different domains and observing communities. The elements defined in this requirement class represent a minimum but with an expectation of community / domain specific requirements to be developed.

Example: ‘QualityFlag’ object showing the schema the flag is from and the flag value. Note that the object has been extended by the inclusion of additional elements providing further context, a description of the test applied and the test name.

|  |
| --- |
| {  "inSchema": "http://www.ukargo.net/data/quality\_control/#realtimeQC",  "flag":"http://www.example.com/QCFlags/good",  "description": "Tests for all salinity or all temperature values in a  profile being the same.",  "test": "Stuck value"  } |

### 251–2024.9 Requirements class: Parameter

251–2024.9.1 A ‘Parameter’ object shall conform with the requirements class specified in Table 251–2024.9.

**Table 251–2024.9: Requirements class /req/parameter**

|  |  |
| --- | --- |
| Requirements class | |
| /req/parameter | |
| Target type | Data instance |
| Name | Parameter |
| Dependency | <https://www.w3.org/TR/prov-dm/> |
| Dependency | <https://www.w3.org/TR/2013/REC-prov-o-20130430/> |
| Dependency | <https://www.w3.org/TR/2013/NOTE-prov-xml-20130430/> |
| Dependency | /req/core/vocabularies |
| Recommendation | /rec/parameter/hasProvenance  A ‘Parameter’ object should provide comprehensive information on the source of the parent observation and any processing applied, such as data transformations, quality control measures, homogenization etc. via a reference to a PROV document using the ‘hasProvenance’ element. |
| Requirement | /req/parameter/hasProvenance/encoding  When available, the ‘hasProvenance’ element shall be encoded as an HTTP URI that resolves to a valid PROV-XML or PROV-O document. |
| Requirement | /req/parameter/status  A ‘Parameter’ object shall provide information on the publication status of the parent observation via the ‘status’ element. |
| Requirement | /req/parameter/status/encoding  The ‘status’ element shall be provided as an HTTP URI that resolves to an online resource providing a definition of the status. |
| Requirement | /req/parameter/version  A ‘Parameter’ object shall provide version information for the parent observation via the ‘version’ element. |
| Requirement | /req/parameter/version/encoding  The ‘version’ element shall be encoded as an integer value and shall be incremented from low to high values with each subsequent version. |
| Permission | /per/parameter/comment  A ‘Parameter’ object may provide additional free text information on the parent observation via the ‘comment’ element. |
| Requirement | /req/parameter/comment/encoding  When provided, the ‘comment’ element shall be encoded as a string in UTF-8. |
| Recommendation | /rec/parameter/reportType  When the parent observation of a ‘Parameter’ object forms part of a wider collection, such as from a synoptic weather report, the report type should be provided via the ‘reportType’ element. |
| Requirement | /req/parameter/reportType/encoding  When the ‘reportType’ is provided it shall be encoded as an HTTP URI reference to a controlled vocabulary or code list entry. |
| Recommendation | /rec/parameter/reportIdentifier  When the parent observation of a ‘Parameter’ object forms part of a wider collection, such as a synoptic weather report, and has been assigned a unique identifier that identifier should be provided via the ‘reportIdentifier’ element. |
| Requirement | /req/parameter/reportIdentifier/encoding  When provided, the ‘reportIdentifier’ shall be encoded as an UTF-8 encoded string. |
| Requirement | /req/parameter/isMemberOf  A ‘Parameter’ object shall provide information on the dataset that the parent observation is a member of via the ‘isMemberOf’ element. |
| Requirement | /req/parameter/isMemberOf/encoding  The ‘isMemberOf’ element shall be encoded as an HTTP URI and shall resolve to a valid WCMP2 metadata record that describes the dataset or collection. |
| Permission | /per/parameter/additionalProperties  A ‘Parameter’ object may provide additional information and metadata associated with the parent observation via the ‘additionalProperties’ element.  Examples include information such as additional contextual metadata or the location or the observation using an alternative Coordinate Reference System. |
| Requirement | /req/parameter/additionalProperties/encoding  When additional information is provided via the ‘additionalProperties’ element the information shall be encoded as key/value pairs. The data type used to represent the values in the pairs shall be appropriate to the additional information provided and each key shall be unique. |

Example: ‘Parameter’ object showing example values of the different elements. As with other examples above, example.com has been used in the HTTP URIs for readability. This example also includes an additional property providing information on the method used to average the data, linking to a method defined in the Manual on codes.

|  |
| --- |
| {  "hasProvenance": "http://data.example.com/prov/1",  "status":"http://vocab.example.com/status/draft",  "version":1,  "comment": "Draft record prior to full QC",  "reportType": "http://vocab.example.com/reportType/daycli",  "reportIdentifier": "0–0-0-MYSTATION\_19931–01\_DAILY",  "isMemberOf": "http://data.example.com/collections/daily\_climate",  "additionalProperties": {  "averagingMethod": "http://codes.wmo.int/bufr4/codeflag/08–094/2"  }  } |

### 251–2024.10 Requirements class: Feature

251–2024.10.1 A 'Feature' object shall conform with the requirements class specified in Table 251–2024.10.

**Table 251–2024.10: Requirements class /req/feature**

|  |  |
| --- | --- |
| Requirements class | |
| /req/feature | |
| Target type | Data instance |
| Name | Feature |
| Requirement | /req/feature/id  The ‘id’ element of a ‘Feature’ object shall be used to link to a definition of the feature. |
| Requirement | /req/feature/id/encoding  The ‘id’ element shall be encoded as an HTTP URI and the URI shall resolve to an online resource providing the definition. |
| Permission | /per/feature/generic  The feature referenced may be a generic feature such as the surface boundary layer or a specific feature such a sampling point. |
| Permission | /per/feature/label  The ‘Feature’ object may provide a human readable representation via the ‘label’ element. |
| Requirement | /req/feature/relation  The relation element shall specify whether the feature referenced is a proximate or ultimate feature of interest and take the value "proximate" or "ultimate" respectively. The value shall be encoded as a UTF-8 string. |

Example: ‘Feature’ object showing the use of an HTTP URI to link to a definition of the surface boundary layer as the ultimate feature of interest.

|  |
| --- |
| {  "id": "http://vocab.example.com/feature/surfaceBoundaryLayer",  "label": "Surface boundary layer",  "relation": "ultimate"  } |

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