

WMO Core Metadata Profile (WCMP) Version 2

World Meteorological Organization

Date: 2023-08-16

Version: 2.0.0-DRAFT-2023-08-16

Document location: TBD

Document status: DRAFT

Task Team on WIS Metadata (TT-WISMD)^[1]

Expert Team on Metadata Standards (ET-Metadata)^[2]

Standing Committee on Information Management and Technology (SC-IMT)^[3]

Commission for Observation, Infrastructure and Information Systems (INFCOM)^[4]

Copyright © 2022 World Meteorological Organization (WMO)

Table of Contents

1. Scope	5
2. Conformance	6
3. References	7
4. Terms and definitions	8
4.1. Abbreviated terms	8
5. Conventions	10
5.1. Identifiers	10
5.2. Examples	10
5.3. Schemas	10
5.4. Schema representation	10
5.4.1. Properties	10
5.5. Use of HTTPS	10
6. Introduction	12
6.1. Overview	12
6.1.1. WIS2	12
6.1.2. Discovery metadata design considerations	13
6.1.3. Granularity	13
6.2. User stories	15
6.2.1. Domain specialist	15
6.2.2. Mass market	15
6.2.3. Developers	15
6.3. OGC API - Records - Part 1: Core	16
6.4. The WIS2 Global Discovery Catalogue	16
6.5. Mass market considerations	16
7. The WMO Core Metadata Profile Record	17
7.1. Conformance Class Core	17
7.1.1. Overview	17
7.1.2. Validation	19
7.1.3. Identifier	19
7.1.4. Conformance	20
7.1.5. Properties / Type	20
7.1.6. Properties / Title	21
7.1.7. Properties / Description	21
7.1.8. Properties / Keywords	22
7.1.9. Properties / Themes	22
7.1.10. Geospatial and Temporal Extents	25
7.1.11. Properties / Contacts	30
7.1.12. Properties / Version	31

7.1.13. Properties / Persistent Identifiers	32
7.1.14. Properties / Record Creation Date	33
7.1.15. Properties / Record Update Date	33
7.1.16. Properties / WMO Data Policy	34
7.1.17. Distribution Information	36
7.1.18. Additional Properties	42
Annex A: Conformance Class Abstract Test Suite (Normative)	43
A.1. Conformance Class: Core	43
A.1.1. Validation	43
A.1.2. Identifier	43
A.1.3. Conformance	44
A.1.4. Type	45
A.1.5. Geospatial Extent	45
A.1.6. Temporal Extent	45
A.1.7. Title	46
A.1.8. Description	46
A.1.9. Themes	47
A.1.10. Contacts	48
A.1.11. Record Creation Date	49
A.1.12. WMO Data Policy	49
A.1.13. Links	50
Annex B: Schemas (Normative)	52
B.1. WMO Core Metadata Profile Schema	52
Annex C: Examples (Informative)	85
C.1. WMO Core Metadata Profile Examples	85
Annex D: Codelists (Normative)	96
Annex E: Bibliography	97
Annex F: Revision History	98

i. Abstract

Discovery and search are key components of WIS2. Quality discovery metadata that leverages industry standards, best practices, and architecture helps lower the barrier to and extend the reach of weather/climate/water data.

This document defines the content, structure, and encoding for the WMO Core Metadata Profile (WCMP). This standard is a profile and extension of the OGC API - Records standard ^[5].

WCMP documents provide descriptive information for discovery of WMO resources. WMO resources include, but are not limited to, data (NWP models, observations, forecasts and warnings, etc.), services/APIs, and processes.

WCMP documents shall be encoded in GeoJSON (RFC 7946 ^[6]) as defined in this specification and shall be made available as HTTP crawlable files or via API provisioning as defined by OGC API - Records.

Weather/climate/water data is by nature geospatial and temporal. The W3C Data on the Web Best Practices ^[7] and Spatial Data on the Web Best Practices ^[8] publications provide guidelines on how to best enable spatiotemporal data to lower the barrier for users, search engine optimization, and linked data. This also aligns with the FAIR data principles (Findable, Accessible, Interoperable, Reusable) ^[9].

ii. Keywords

The following are keywords to be used by search engines and document catalogues.

wmo, wis, wis2, weather, climate, water, metadata, discovery, search

iii. Security Considerations

Based on the WMO Unified Data Policy for the International Exchange of Earth System Data (Resolution 1 (Cg-Ext(2021)) ^[10], exchanged data are classified as **core** or **recommended**. Core data is considered fully open and unrestricted with no security considerations. Recommended data may have access control defined.

No additional security considerations have been made for this standard.

Chapter 1. Scope

This document defines the content, structure, and encoding of discovery metadata published as part of the WIS2 Global Discovery Catalogue (GDC).

The WMO Core Metadata Profile (WCMP) standard defined herein is an extension of the International Standard *OGC API - Records - Part 1: Core*.

WCMP discovery metadata records shall be encoded as GeoJSON as defined in *OGC API - Records - Part 1: Core*.

The primary purpose of WCMP is to describe **datasets**. While WCMP can describe any resource (datasets, APIs and data reduction services, processes, analytics, etc.), the key component for international data exchange is data. Services and APIs operating on resources may be represented as part of their own metadata or associated through WCMP links.

WCMP discovery metadata record provides descriptions at the granularity level of a dataset. Station, instrument and observation metadata are supported by the *WIGOS Metadata Standard (WMO-No. 1192)* ^[11].

This specification defines the conformance requirements for the WMO Core Metadata Profile. Annex A defines the abstract test suite. Annex B provides normative information on schemas. Annex C provides informative examples. Annex D provides normative information on codelists.

[1] <https://community.wmo.int/governance/commission-membership/commission-observation-infrastructures-and-information-systems-infcom/commission-infrastructure-officers/infcom-management-group/standing-committee-information-management-and-technology-sc-int/expert-team-metadata-0>

[2] <https://community.wmo.int/governance/commission-membership/commission-observation-infrastructures-and-information-systems-infcom/commission-infrastructure-national-representatives/infcom-management-group/standing-committee-information-management-and-technology-sc-int/et-metadata>

[3] <https://community.wmo.int/governance/commission-membership/commission-observation-infrastructures-and-information-systems-infcom/commission-infrastructure-officers/infcom-management-group/standing-committee-information-management-and-technology-sc-int>

[4] <https://community.wmo.int/governance/commission-membership/infcom>

[5] <https://ogcapi.ogc.org/records>

[6] <https://datatracker.ietf.org/doc/html/rfc7946>

[7] <https://www.w3.org/TR/dwbp>

[8] <https://www.w3.org/TR/sdw-bp>

[9] https://en.wikipedia.org/wiki/FAIR_data

[10] https://library.wmo.int/doc_num.php?explnum_id=11113#page=9

[11] https://library.wmo.int/doc_num.php?explnum_id=10109

Chapter 2. Conformance

Conformance with this standard shall be checked using the tests specified in Annex A (normative) of this document.

OGC API - Records (OARec) provides a record metadata model in support of resource discovery. This standard is an extension of *OGC API - Records - Part 1: Core*. Conformance to this standard requires demonstrated conformance to the applicable Conformance Classes of *OGC API - Records - Part 1: Core*.

Authors of discovery metadata records published within the WIS2 Global Discovery Catalogue (GDC) are required to comply with the WMO Core Metadata Profile (WCMP). WCMP discovery metadata shall therefore be compliant with OGC API - Records - Part 1: Core: Requirements Class: Record Core.

WMO shall publish guidance material to assist authors of WCMP discovery metadata for maintaining consistency across multiple metadata records and perform quality assessment and reporting.

This standard identifies one Conformance Class which defines the functional requirements.

The mandatory Conformance Class for WCMP is:

- "WMO Core Metadata Profile Core": This conformance class inherits from *OGC API — Records — Part 1: Core: Requirements Class: Record Core* which defines the requirements for a catalogue record. The requirements specified in the Requirements Class "Record Core" are mandatory for all implementations of WMCP. The requirements are specified in Chapter 7 and in Annex A in more detail.

Chapter 3. References

- OGC: OGC 20-004, OGC API - Records - Part 1: Core 1.0 (2023) ^[12]
- OGC: OGC 17-069r, OGC API - Features - Part 1: Core 1.0 (2022) ^[13]
- IETF: RFC-7946 The GeoJSON Format (2016) ^[14]
- IETF: RFC-8259 The JavaScript Object Notation (JSON) Data Interchange Format the GeoJSON Format (2016) ^[15]
- W3C/OGC: Spatial Data on the Web Best Practices, W3C Working Group Note (2017) ^[16]
- W3C: Data on the Web Best Practices, W3C Recommendation (2017) ^[17]
- W3C: Data Catalog Vocabulary, W3C Recommendation (2014) ^[18]
- IANA: Link Relation Types (2020) ^[19]
- Linux Foundation: SPDX License List (2021) ^[20]
- IETF: JSON Schema (2022) ^[21]
- WMO: WIS2 Topic Hierarchy (2022) ^[22]
- OpenAPI Specification 3.1.0 (2022) ^[23]

[12] <https://docs.ogc.org/DRAFTS/20-004.html>

[13] <https://docs.openeospatial.org/is/17-069r4/17-069r4.html>

[14] <https://datatracker.ietf.org/doc/html/rfc7946>

[15] <https://datatracker.ietf.org/doc/html/rfc8259>

[16] <https://www.w3.org/TR/sdw-bp>

[17] <https://www.w3.org/TR/dwbp>

[18] <https://www.w3.org/TR/vocab-dcat>

[19] <https://www.iana.org/assignments/link-relations/link-relations.xml>

[20] <https://spdx.org/licenses>

[21] <https://json-schema.org>

[22] <https://github.com/wmo-im/wis2-topic-hierarchy>

[23] <https://github.com/OAI/OpenAPI-Specification/blob/3.1.0/versions/3.1.0.md>

Chapter 4. Terms and definitions

This document uses the terms defined in [OGC Policy Directive 49](#), which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this Standard and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications ([OGC 08-131r3](#)), also known as the 'ModSpec'. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

The following additional terms and definitions also apply.

4.1. Abbreviated terms

Table 1. Symbols and abbreviated terms

Abbreviation	Term
API	Application Programming Interface
ARK	Archival Resource Key
DCAT	Data Catalog Vocabulary
DCPC	Data Collection and Production Centres
DOI	Digital Object Identifier
GDC	Global Discovery Catalogue
GIS	Geographic Information System
GISC	Global Information System Centre
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
NC	National Centre
NWP	Numerical Weather Prediction
OARec	OGC API - Records
OGC	Open Geospatial Consortium
REST	Representational State Transfer

Abbreviation	Term
ROA	Resource-oriented architecture
S3	Simple Storage Service
SEO	Search engine optimization
SOA	Service-oriented architecture
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WCMP	WMO Core Metadata Profile
WIS	WMO Information System
WMO	World Meteorological Organization
XML	eXtensible Markup Language

Chapter 5. Conventions

This section provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of JSON schema, or special notes regarding how to read the document.

5.1. Identifiers

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

<http://wis.wmo.int/spec/wcmp/2>

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

5.2. Examples

Examples provided in this specification are encoded as GeoJSON.

Complete examples can be found at <https://schemas.wmo.int/wcmp/2.0/examples>

5.3. Schemas

The WCMP schema can be found at <https://schemas.wmo.int/wcmp/2.0/wcmp2-bundled.json>

5.4. Schema representation

JSON Schema ^[24] objects are used throughout this standard to define the structure of metadata records. These schema objects are also typically represented using YAML ^[25]. YAML is a superset of JSON, and in this standard are regarded as equivalent.

Metadata record instances are always represented as JSON.

5.4.1. Properties

A JSON **property** represents a key-value pair, where the key is the name of the property and the value is a standard JSON data type.

```
"myPropertyName": "test123"
```

5.5. Use of HTTPS

For simplicity, this document only refers to the HTTP protocol. This is not meant to exclude the use of HTTPS and simply is a shorthand notation for "HTTP or HTTPS." In fact, most servers are expected to use HTTPS, not HTTP.

[24] <https://json-schema.org>

[25] <https://en.wikipedia.org/wiki/YAML>

Chapter 6. Introduction

6.1. Overview

The initial WIS technical specifications were developed using service-oriented architecture (SOA) principles. It featured complex ISO 19115/19139-based XML for metadata. Since then, international standards development organizations such as W3C and OGC have moved to adopt a more resource-oriented architecture (ROA), leveraging RESTful design patterns, and mass market encodings such as JSON and HTML.

Aligning with the WIS2 Principles, in order to support the WIS2 Technical Specifications for discovery and search, discovery metadata will be published to a global discovery catalogue, which will provide an OGC API - Records searchable functionality. Users will be able to search from a web browser, whereas machines will interact with an API.

6.1.1. WIS2

WIS2 puts forth the following principles (those focused on discovery metadata are in **bold**):

- **(1) WIS 2.0 adopts Web technology and leverages industry best practices and open standards.**
- **(2) WIS 2.0 uses Uniform Resource Locators (URL) to identify resources (i.e., Web pages, data, metadata, APIs) use.**
- **(3) WIS 2.0 prioritizes the use of public telecommunications networks (i.e. Internet) when publishing digital resources.**
- **(4) WIS 2.0 requires provision of Web service(s) to access or interact with digital resources (e.g. data, information, products) published using WIS.**
- (5) WIS 2.0 encourages NCs and DCPCs to provide 'data reduction' services via WIS that process 'big data' to create results or products that are small enough to be conveniently downloaded and used by those with minimal technical infrastructure.
- **(6) WIS 2.0 adds open standard messaging protocols that use the publish-subscribe message pattern to the list of data exchange mechanisms approved for use within WIS and GTS.**
- (7) WIS 2.0 requires all services that provide real-time distribution of messages (containing data or notifications about data availability) to cache/store the messages for a minimum of 24 hours and allow users to request cached messages for download.
- (8) WIS 2.0 adopts direct data-exchange between provider and consumer and phases out the use of routing tables and bulletin headers.
- **(9) WIS 2.0 provides a catalogue containing metadata that describes both data and the service(s) provided to access that data.**
- **(10) WIS 2.0 encourages data providers to publish metadata describing their data and Web services in a way that can be indexed by commercial search engines.**

6.1.2. Discovery metadata design considerations

The following describes envisioned workflows of WIS2 in the context of metadata search and harvesting.

- Flexible metadata publishing mechanisms so providers can publish discovery metadata in the easiest and most efficient way possible.
- Basic, HTTP crawlable metadata files (filesystem, object storage). For example, publishing discovery metadata as JSON files to an S3 bucket, and then making that bucket available for harvesting and traversal to search engines and metadata harvesters.
- The browser as the catalogue. Browsers utilize mass market search engines as the gateway to low barrier discovery. This pattern may work by way of the Global Discovery Catalogue and the Global Cache, with the idea that search engines can index from the authoritative source.

6.1.3. Granularity

In order to provide discovery metadata of value, it is important to clarify the granularity levels of which providers are to provide describing their data and associated services. Articulating the level of granularity will reduce catalogue "pollution" and bring the user closer to the data via their search criteria.

A WCMP record provides a description at the primary granularity of a **dataset**, which facilitates clear cataloguing and discovery workflow, in combination with data services or APIs, which provide access, queries, and filters at a lower level of granularity (parameter, variable, spatiotemporal extents).

The [Discovery metadata workflow](#) figure below illustrates example metadata publication and discovery workflows against common meteorological data types:

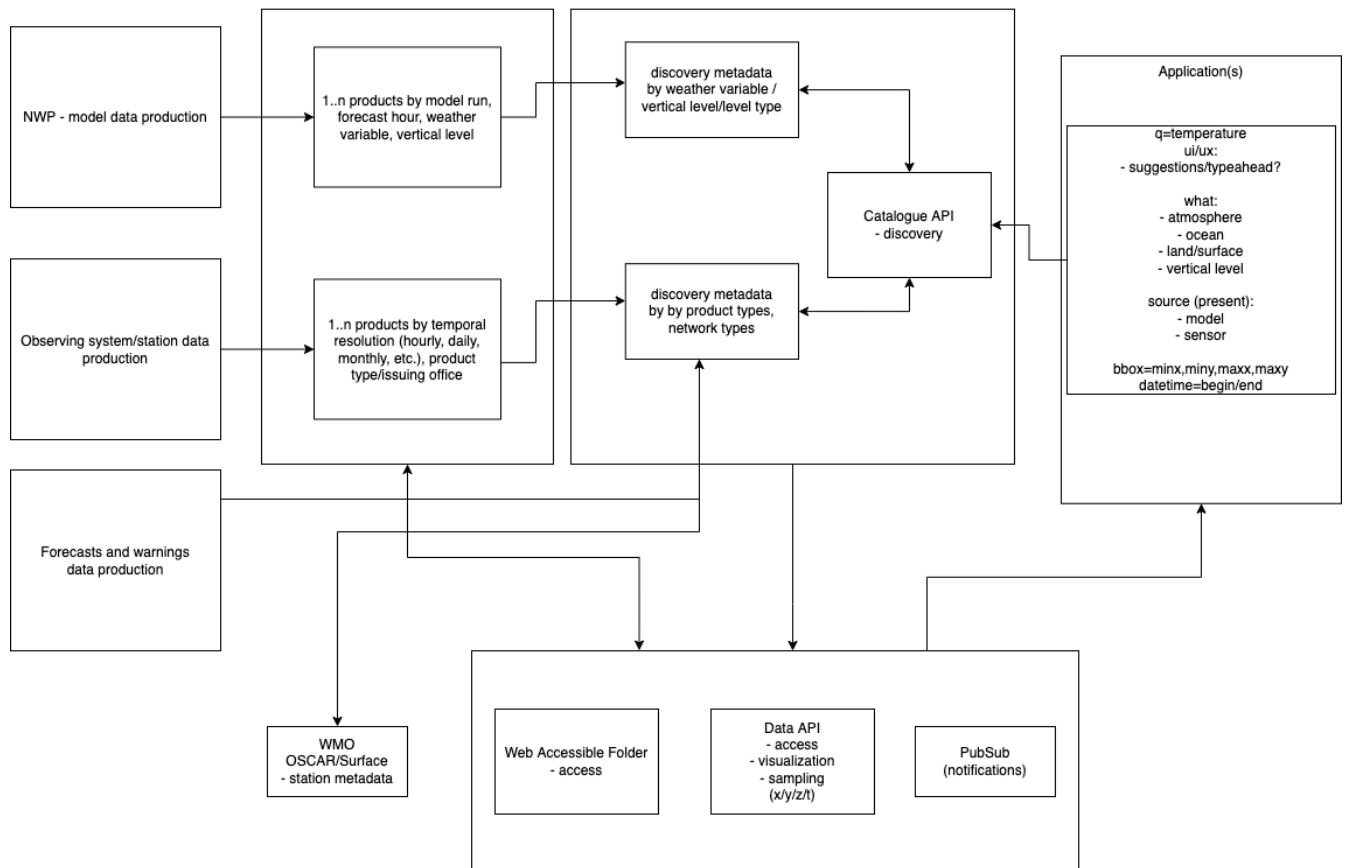


Figure 1. Discovery metadata workflow

Common meteorological data types include:

- collection (model): NWP model (discovery metadata record). Example: Canadian Global Deterministic Prediction System
 - variables are described as themes/concepts (e.g. air temperature)
 - access options:
 - data service or API endpoint to interrogate data/variable/spatiotemporal dimensions
 - x/y/z/t (granule) (STAC Item with link to actual data asset)
- collection (observations): surface weather observations (discovery metadata record)
 - station metadata as WIGOS metadata via WMO OSCAR/Surface
 - access options:
 - data service or API endpoint to interrogate data/variable/spatiotemporal dimensions
 - x/y/z/t (granule) (STAC Item, with link to actual data asset)
- collection (climate archive): climate daily station data (discovery metadata record)
 - access options:
 - API endpoint to interrogate data/variable/spatiotemporal dimensions
 - single observation (granule) (STAC Item with link to actual data asset)

6.2. User stories

As part of requirements gathering ^[26], the following user stories provide a description of features that are relevant to WIS2 metadata and search, and are cast from a user perspective:

6.2.1. Domain specialist

- As an NWP centre operator, I want to quickly and easily publish information about the data that my centre provides and update it as needed in a (semi)automated way using the information that I already have in my databases, so that I can concentrate on my core business.
- As the leader of a forecasting team of a national meteorological institution, I would like to be able to find more sources of data that might be relevant or useful for the work of my team, notably NWP and satellite imagery so that we could further improve our predictions.
- As a GIS professional, I would like to search for weather, climate and water data from my GIS desktop support tool so that I can integrate forecast data into my workflow.

6.2.2. Mass market

- As an entrepreneur (start-up) that provides (wants to provide) tailored weather information, I want to find services (free or commercial) that provide meteorological data in the cloud, or even better, a service that provides customizable processing of such data to build my own service on top of it. I also want to be able to find out if a new service becomes available or if an existing one changes its abilities so that my company can keep on advancing.
- As a user, I would like to search for real-time observations for a given time and geographical area of interest, so that I can have up-to-date information on weather for my city.

6.2.3. Developers

- As a software developer (working for a national meteorological centre or a private company), I would like to find a relevant technical description of the service (API) that my boss wants me to integrate with, so that the declared interoperability becomes a reality.
- As a web developer, I would like to access a search API that provides easy-to-read documentation, examples, and a simple, intuitive RESTful API with JSON so that I can integrate into my web application quickly.

The following WIS2 marketing video ^[27] adds the following user stories:

- As an everyday user, I would like to find easy-to-understand and precise weather data so that I can plan to have people over for an outdoor BBQ on a nice day.
- As a smart home owner, I would like access to frequently updated data so that I can keep my smart home monitoring up to date.
- As a weather specialist, I would like to access weather data in native data formats and subscribe to data updates, so that I can provide tailor-made weather services to my users.

Given the above, we see a variety of users/actors to which WIS2, driving the need for a low barrier, ubiquitous and efficient discovery, visualization, and access of weather, climate, water (real-time,

near real-time, archive, etc.) data.

6.3. OGC API - Records - Part 1: Core

The OGC Records - API - Part 1: Core specification:

- lowers the discovery barrier to finding the existence of geospatial resources on the Web
- provides the ability for discovery metadata to be published via API machinery (searchable catalogue) or static records (crawlable catalogue)
- provides a core record model for information communities to extend
- provides a subset of core queryables (e.g. by resource type, by external identifier) which enables federation and cross catalogue discovery functionality

6.4. The WIS2 Global Discovery Catalogue

The GDC will provide a central search endpoint, enabling users to traverse, browse and search data holdings in WIS2. Key search predicate capabilities include:

- geospatial (**bbox=**)
- temporal (time instant or time period) (**datetime=**)
- equality predicates (i.e. **property=value**) for any defined discovery metadata property
- full-text (**q=**)

Given the WIS2 principles, use cases, OGC API - Records - Part 1: Core, and the WIS2 Global Discovery Catalogue, WCMP provides a standards-based, clear and well-defined information model to facilitate the management and discovery of data within WIS2.

6.5. Mass market considerations

Given WIS2 principle 10 (publishing metadata in a way that commercial search engines can index), WCMP discovery metadata enables annotations that can facilitate Search Engine Optimization (SEO) and structured data discovery, search, and relevant results.

[26] <https://github.com/wmo-im/wcmp/issues/107>

[27] https://gisc.dwd.de/wis2.0/WIS_2.0_final.mp4

Chapter 7. The WMO Core Metadata Profile Record

A WCMP record provides descriptive information about a dataset made available through WIS2.

7.1. Conformance Class Core

7.1.1. Overview

The WCMP Core Conformance Class provides requirements to articulate the required elements of a WCMP discovery metadata record.

Requirements Class	
http://www.wmo.int/spec/wcmp/2/conf/core	
Target type	Discovery Metadata
Dependency	IETF RFC 8259: The JavaScript Object Notation (JSON) Data Interchange Format
Dependency	JSON Schema
Dependency	GeoJSON
Dependency	OGC API - Records - Core: Part 1
Dependency	OpenAPI Specification, Version 3.1.0
Pre-conditions	The record conforms to OGC API - Records - Core: Part 1: Requirements Class: Record Core

The table below provides an overview of the set of properties that may be included in a WCMP record.

Table 2. WCMP record core properties

Property	Requirement	Description
<code>id</code>	required	A unique identifier to the dataset (see Identifier)
<code>type</code>	required	A fixed value denoting the record as a GeoJSON Feature (see Validation)
<code>conformsTo</code>	required	The version of WCMP associated that the record conforms to (see Conformance)
<code>properties.type</code>	required	The resource type described by the record (see Properties / Type)

Property	Requirement	Description
<code>properties.title</code>	required	A human-readable name for a given dataset (see Properties / Title)
<code>properties.description</code>	required	A free-text summary description of the dataset (see Properties / Description)
<code>properties.keywords</code>	optional	Keywords, tags, key phrases, or classification codes (see Properties / Keywords)
<code>properties.themes</code>	required	Classifiers, categories and controlled vocabularies (see Properties / Themes)
<code>geometry</code>	required	Geospatial location associated with the dataset, in a geographic coordinate reference system (see Geospatial Extent)
<code>time</code>	required	Temporal extent associated with a dataset (see Temporal Extent)
<code>additionalExtents.spatial</code>	optional	Additional geospatial extents in other coordinate reference systems (see Additional Geospatial Extents)
<code>additionalExtents.temporal</code>	optional	Additional time instants or periods (see Additional Temporal Extents)
<code>properties.contacts</code>	required	Contact information for the dataset (see Properties / Contacts)
<code>properties.version</code>	optional	Version or edition of the dataset (see Properties / Version)
<code>properties.externalIds</code>	optional	Persistent identifiers or handles for the dataset (see Properties / Persistent Identifiers)
<code>properties.created</code>	required	The date that the record was created (see Properties / Record Creation Date)
<code>properties.updated</code>	optional	The date that the record was updated (see Properties / Record Update Date)

Property	Requirement	Description
<code>properties.wmo:dataPolicy</code>	conditional	WMO data policy as per Resolution 1 (Cg-Ext(2021)) ^[28] . Required for datasets (see Properties / WMO Data Policy)
<code>links</code>	required	Online linkages to data retrieval or additional resources associated with the dataset (see Distribution Information)
<code>properties.*</code>	optional	Additional properties as required (see Additional Properties)

7.1.2. Validation

The WCMP record schema is based on *OGC API - Records - Part 1: Core: Requirements Class: Record Core* schema and the associated information model. WCMP records compliant with WCMP schema are therefore compliant with the *OGC API - Records* record schema.

Requirement 1	/req/core/validation
A	Each WCMP record SHALL validate without error against the WCMP schema.
B	Each WCMP record SHALL provide <code>id</code> , <code>type</code> , <code>geometry</code> and <code>properties</code> properties for GeoJSON compliance.
C	Each WCMP record <code>type</code> property SHALL be set to a fixed value of <code>Feature</code> for GeoJSON compliance.

7.1.3. Identifier

The `id` property is a unique identifier of the dataset. A record identifier is essential for querying and identifying records within the GDC.

Requirement 2	/req/core/identifier
A	A WCMP record SHALL have an identifier via the <code>id</code> property.
B	A WCMP record identifier SHALL have the following notation: <code>urn:x-wmo:md:{country}:{centre_id}:{local_identifier}</code> .
C	A WCMP record identifier SHALL include a <code>country</code> based on the associated vocabulary as specified in the WIS topic hierarchy .

D	A WCMP record identifier SHALL include a centre_id based on the associated vocabulary specified in the WIS topic hierarchy .
E	A WCMP record identifier SHALL include a local identifier as defined by the data publisher. The local identifier SHALL NOT have spaces or special or accented characters.

Example

```
"id": "urn:x-wmo:md:can:eccc-msc:observations.swob"
```

7.1.4. Conformance

The **conformsTo** property identifies the version of the WCMP standard that the metadata record conforms to. Conformance identification is valuable for version detection and handling of content.

Requirement 3	/req/core/conformance
A	A WCMP record SHALL provide information on conformance via the OAREc record conformsTo property.
B	A WCMP record conformsTo property SHALL advertise conformance to WCMP.

Example

```
"conformsTo": [
  "http://wis.wmo.int/spec/wcmp/2/conf/core"
]
```

7.1.5. Properties / Type

The **type** property identifies the type of resource that the metadata record describes using values from the resource type codelist. While most metadata records will be set to **dataset**, WCMP can also describe services and processes. See the resource type codelist ^[29] for allowed values.

Requirement 4	/req/core/type
A	A WCMP record SHALL provide a properties.type property.
B	A WCMP record's properties.type property SHALL provide a valid resource type from the WCMP resource type codelist.

Requirement 5	/req/core/type_service
A	A WCMP record describing a service or API SHALL set properties.type to service .

Example

```
"properties": {
  ...
  "type": "dataset"
  ...
}
```

7.1.6. Properties / Title

The **title** property is a human-readable name of the dataset.

Requirement 6	/req/core/title
A	A WCMP record SHALL provide a properties.title property.

Example

```
"properties": {
  ...
  "title": "Surface weather observations"
  ...
}
```

7.1.7. Properties / Description

The **description** property is a free-text summary of the dataset.

Table 3. Example

Requirement 7	/req/core/description
A	A WCMP record SHALL provide a properties.description property.

```

"properties": {
  ...
  "description": "Surface observations measured at the automatic and manual stations
of the Environment and Climate Change Canada and partners networks, either for a
single station or for the stations of specific provinces and territories (last 30
days)."
```

7.1.8. Properties / Keywords

The **keywords** property is a list keywords, tags, specific phrases, or classification codes associated with the resource, but are not referenced to a particular vocabulary or knowledge organization system.

Recommendation 1	/rec/core/keywords
A	A WCMP record SHOULD provide a properties.keywords property, as a list of freeform text or tags that are not based on a controlled vocabulary.

Example

```

"properties": {
  ...
  "keywords": [
    "surface",
    "observations",
    "weather",
    "real-time"
  ]
  ...
}
```

7.1.9. Properties / Themes

The **themes** property is a list of concepts that are referenced to a vocabulary or knowledge organization system used to classify the resource. A concept's **id** provides an identifier for the given concept, as well as optional **title**, **description** and **url** properties to further describe the concept as desired.

A theme's **scheme** property provides a remote reference to a controlled vocabulary, codelist or knowledge organization system.

A concept's optional **url** property can provide a remote reference to a given concept (for example, to an online ontology or code registry).

A WCMP metadata record can have one or more themes.

Requirement 8	/req/core/themes
A	A WCMP record SHALL use themes/concepts when describing a dataset based on an associated controlled vocabulary.
B	A WCMP record SHALL provide at least one theme.
C	Within each theme, a WCMP record SHALL provide at least one concept.
D	Within each theme, a WCMP record SHALL provide a scheme that refers to a controlled vocabulary or thesaurus.
E	A WCMP record SHALL provide a theme/concept identifying all applicable Earth system disciplines as defined in WCMP2 codelists.

Requirement 9	/req/core/themes_wis2_global_service
A	A WCMP record describing a WIS2 global service SHALL provide a theme/concept identifying all Earth system disciplines as defined in WCMP2 codelists.
B	A WCMP record describing a WIS2 global service SHALL provide a theme/concept identifying the service type provided.

Recommendation 2	/rec/core/granularity
A	A WCMP record SHOULD describe dataset parameters and variables as themes/concepts with an associated controlled vocabulary.

Examples

```
"properties": {  
  ...  
  "themes": [  
    {  
      "concepts": [  
        {  
          "id": "Temperature",  
          "url": "http://codes.wmo.int/grib2/codeflag/4.2/_0-0-0"  
        },  
        {  
          "id": "Relative humidity",  
          "url": "http://codes.wmo.int/grib2/codeflag/4.2/_0-1-1"  
        },  
        {  
          "id": "u-component of Wind",  
          "url": "http://codes.wmo.int/grib2/codeflag/4.2/_0-2-2"  
        },  
        {  
          "id": "v-component of Wind",  
          "url": "http://codes.wmo.int/grib2/codeflag/4.2/_0-2-3"  
        }  
      ],  
      "scheme": "http://codes.wmo.int/grib2/codeflag/4.2"  
    }  
  ]  
  ...  
}
```

```
"themes": [  
  {  
    "concepts": [  
      {  
        "id": "Temperature/air temperature",  
        "url": "http://codes.wmo.int/bufr4/b/12/_001"  
      },  
      {  
        "id": "Dewpoint temperature",  
        "url": "http://codes.wmo.int/bufr4/b/12/_003"  
      }  
    ],  
    "scheme": "http://codes.wmo.int/bufr4/b"  
  }  
]
```

```

"properties": {
  ...
  "themes": [
    {
      "concepts": [{
        "id": "weather",
        "url": "https://github.com/wmo-im/wcmp2-codelists/blob/main/codelists/earth-
system-domain.csv#L2"
      }],
      "scheme": "https://github.com/wmo-im/wcmp2-codelists/blob/main/codelists/earth-
system-domain.csv"
    }
  ]
  ...
}

```

7.1.10. Geospatial and Temporal Extents

7.1.10.1. Geospatial Extent

The **geometry** property is the general bounding spatial extent of the dataset in the geographic coordinate system. Geospatial bounding extents provide a useful indicator of the location of the dataset to facilitate search and map displays in the GDC.

Requirement 10	/req/core/extent_geospatial
A	A WCMP record SHALL provide ONE geometry property to convey the geospatial properties of a dataset using a geographic coordinate reference system (World Geodetic System 1984 [WGS 84]) and longitude and latitude decimal degree units.
B	A WCMP record SHALL provide geometry values as integer or float data types.
C	A WCMP record SHALL provide the value of null when a conformant geometry cannot be derived.

Recommendation 3	/rec/core/extent_geospatial_point
A	For datasets based on a geometry without a calculated area (i.e. single station point), a WCMP record SHOULD provide the GeoJSON geometry as a Point type.

Recommendation 4	/rec/core/extent_geospatial_precision
------------------	---------------------------------------

A	Geometry coordinates SHOULD have a level of precision of 2 decimal places.
Recommendation 5	/rec/core/extent_geospatial_wis_2_global_service
A	For a WIS2 Global Service, a WCMP record SHOULD provide the GeoJSON geometry as a Polygon type with a WGS84 bounding geometry of -180 (west longitude), -90 (south latitude), 180 (east longitude), 90 (north latitude).

Examples

```
"geometry": {
  "type": "Polygon",
  "coordinates": [[
    [-142.23, 28.03],
    [-142.23, 82.56],
    [-52.16, 82.56],
    [-52.16, 28.03],
    [-142.23, 28.03]
  ]]
}
```

```
"geometry": {
  "type": "Point",
  "coordinates": [-79.38, 43.65]
}
```

```
"geometry": null
```

7.1.10.2. Additional Geospatial Extents

The `additionalExtents.spatial` property is for other geospatial extents associated with the dataset, for example, non-geographic data or for describing multiple sub-areas of a dataset. The `additionalExtents.spatial.bbox` property provides the ability to list one to many minimum bounding geometries for a given dataset, where:

- minx is the westernmost coordinate of the limit of the dataset extent, expressed in longitude decimal degrees as a signed number between -180 and 180, less than or equal to maxx.
- miny is the southernmost coordinate of the limit of the dataset extent, expressed in latitude decimal degrees as a signed number between -90 and 90, less than or equal to maxy.
- maxx is the easternmost coordinate of the limit of the dataset extent, expressed in longitude decimal degrees as a signed number between -180 and 180, greater than or equal to minx.

- maxy is the northernmost coordinate of the limit of the dataset extent, expressed in latitude decimal degrees as a signed number between -90 and 90, greater than or equal to miny.

This property also facilitates catalogue searches with geospatial predicate (within extent, etc.).

Permission 1	/per/core/extent_geospatial
A	A WCMP record MAY provide multiple additionalExtents.spatial.bbox item properties to express additional geospatial extents in other coordinate reference systems.

Example

```
"additionalExtents": {
  "spatial": {
    "bbox": [
      [-142, 42, -52, 84]
    ],
    "crs": "http://www.opengis.net/def/crs/OGC/1.3/CRS84"
  }
}
```

7.1.10.3. Temporal Extent

The **time** property describes the general bounding extent of the dataset and the temporal resolution.

Temporal extents provide a useful indicator of the date and time period of the dataset and facilitates temporal searching in the GDC.

Temporal resolution provides a useful indicator of the data update frequency (for example, for real-time datasets).

Requirement 11	/req/core/extent_temporal
A	A WCMP record SHALL provide ONE time item property using the Gregorian calendar.
B	A WCMP record SHALL provide the value of null when a conformant time cannot be derived.
C	All non-null time values SHALL be valid ISO8601 representations or .. for an open-ended extent.

Recommendation 6	/rec/core/extent_temporal
------------------	---------------------------

A	For datasets with known or discrete intervals, a WCMP record SHOULD provide the temporal resolution (<code>time.resolution</code>) as a valid ISO 8601 duration (e.g., <code>P1D</code>).
B	For complex temporal extents, a WCMP record SHOULD also provide a human-readable explanation in <code>properties.description</code> to promote clarification and unambiguity.
C	To distinguish maintenance or update frequency, those SHOULD be included in <code>properties.themes</code> .

Recommendation 7	/rec/core/extent_service
A	For WCMP records describing a service or API, a <code>time</code> value of <code>null</code> MAY be provided.

Examples

```
"time": {
  "date": "2021-10-30"
}
```

```
"time": {
  "timestamp": "2021-10-30T11:11:11Z"
}
```

```
"time": {
  "interval" : ["2020-10-30", "2021-10-30"],
  "resolution": "P1D"
}
```

```
"time": {
  "interval": [
    ["R/T00Z", "PT180H"],
    ["R/T12Z", "PT180H"]
  ],
  "resolution": "PT6H"
}
```

```
"time": null
```

If only times are given for the interval, it is implicitly assumed that those are recurring every day.

```
"time": {
  "interval" : ["T00Z", "T23Z"],
  "resolution": "PT1H"
}
```

Some cases might be more complex due to the product's characteristics.

The example below describes two intervals with the model run time (start) up to the range covered by the model run (end). Here, the end of the interval is characterised by a time period. Additionally, the time steps within the model range are relevant information (resolution).

To ensure clarity, an additional human-readable explanation is given in the `properties.description`. Further time-related characteristics (e.g. frequency of modifications, available data, etc.) should be added in `properties.themes`.

Example: complex time information with additional explanation and characteristics

```
"time": {
  "interval": [
    ["T00Z", "PT180H"],
    ["T12Z", "PT180H"]
  ],
  "resolution": "PT6H"
},
"properties": {
  ...
  "description": "[...] up to +180h every 6h, runs 00/12 UTC [...].",
  ...
  "themes": [
    {
      "concepts": [
        {"id": "continual"}
      ],
      "scheme":
        "https://standards.iso.org/iso/19139/resources/gmxCodeLists.xml#MD_FrequencyCode"
    }
    ...
  ]
  ...
}
```

If necessary, further granularity can be indicated by the `additionalExtents.temporal` property.

7.1.10.4. Additional Temporal Extents

The `additionalExtents.temporal` property is for describing other temporal extents associated with the dataset, for example, multiple time instances or to identify other temporal reference systems.

Temporal extents can be fully bound or open in either direction (e.g., until or before). An

`additionalExtents.temporal.resolution` property is able express the temporal granularity at which a given dataset's data is measured, generated, or reported.

Permission 2	/per/core/extent_temporal
A	A WCMP record MAY provide multiple <code>additionalExtents.temporal</code> item properties to express additional temporal extents in other temporal reference systems.

Example

```
"additionalExtents": {
  "temporal": {
    "interval": [
      [
        "1924-08-18",
        null
      ]
    ],
    "resolution": "P1H",
    "trs": "http://www.opengis.net/def/uom/ISO-8601/0/Gregorian"
  }
}
```

7.1.11. Properties / Contacts

The `contacts` property is the information associated with one or more responsible parties of the resource.

Requirement 12	/req/core/contacts
A	A WCMP record SHALL provide at least one contact via the <code>properties.contacts</code> property.
B	A WCMP record's contact SHALL provide a <code>organization</code> property.
C	A WCMP record's contact <code>roles</code> , when specified, SHALL provide a role type from the WCMP resource type codelist.

Permission 3	/per/core/contacts
A	A WCMP record <code>properties.contacts</code> property MAY provide more than one contact via multiple objects or a single contact object with multiple roles.

Examples

```
"properties": {
  ...
  "contacts": [{
    "identifier": "ECCC",
    "organization": "Government of Canada; Environment and Climate Change Canada;
Meteorological Service of Canada",
    "positionName": "National Inquiry Response Team",
    "phones": [{
      "value": "+18199972800"
    }],
    "emails": [{
      "value": "enviroinfo@ec.gc.ca"
    }],
    "addresses": [{
      "deliveryPoint": "77 Westmorland Street, suite 260",
      "city": "Fredericton",
      "administrativeArea": "NB",
      "postalCode": "E3B 6Z4",
      "country": "Canada"
    }],
    "links": [{
      "href": "https://www.canada.ca/en/environment-climate-change.html"
    }],
    "contactInstructions": "email",
    "roles": ["producer", "host"]
  }]
  ...
}
```

Note that a contact can also be specified by using a URL:

```
"properties": {
  ...
  "contacts": [{
    "name": "Government of Canada; Environment and Climate Change Canada;
Meteorological Service of Canada",
    "links": [{
      "href": "https://www.canada.ca/en/environment-climate-
change/corporate/contact.html"
    }]
  ]
  ...
}
```

7.1.12. Properties / Version

The **version** property is the version of the dataset. Datasets may be versioned by an organization,

for example, the version of an NWP model or a processing chain/workflow, and data providers can make this information available when there are multiple versions of a dataset over time.

Permission 4	/per/core/version
A	A WCMP record MAY provide a <code>properties.version</code> property to describe the version of a given dataset.

Example

```
"properties": {
  ...
  "version": "0.1.0"
  ...
}
```

7.1.13. Properties / Persistent Identifiers

The `externalIds` property is a persistent (or handle) identifier used to provide a long lasting reference to a digital resource. Persistent identifiers are commonly used for scientific publications and datasets.

Examples of persistent identifiers include, but are not limited to:

- Digital Object Identifiers (DOI) ^[30]
- Archival Resource Key (ARK) ^[31]
- Handle ^[32]

Recommendation 8	/rec/core/pids
A	A WCMP record SHOULD provide persistent identifier references via items in the <code>properties.externalIds</code> array property, where the value of <code>scheme</code> is based on an established persistent identifier scheme (such as <code>doi</code> , <code>ark</code> , <code>handle</code> , etc.), and the value of <code>value</code> is the persistent identifier (e.g. https://dx.doi.org/10.14287/10000001).

Permission 5	/per/core/pids
A	A WCMP record MAY provide a persistent identifier as a means to cite research or resource identification using a persistent identifier scheme/framework.

Example

```
"properties": {
  ...
  "externalIds": [{
    "scheme": "doi",
    "value": "https://dx.doi.org/10.14287/10000001"
  }, {
    "scheme": "hdl",
    "value": "2381/12775"
  }, {
    "scheme": "ark",
    "value": "ark:/13030/tf5p30086k"
  } ]
  ...
}
```

7.1.14. Properties / Record Creation Date

The **created** property is a single date that the WCMP metadata record was created.

Requirement 13	/req/core/record_creation_date
A	A WCMP record SHALL provide a single properties.created property.
B	A WCMP record SHALL NOT be repeatable or used to document change history.

Example

```
"properties": {
  ...
  "created": "2021-06-12T23:45:24Z"
  ...
}
```

7.1.15. Properties / Record Update Date

The **updated** property is the date that the WCMP metadata record was last updated.

Recommendation 9	/rec/core/record_update_date
A	A WCMP record SHOULD provide a properties.updated property when a record has been updated since its initial creation.

Example

```
"properties": {  
  ...  
  "updated": "2022-06-12T18:52:39Z"  
  ...  
}
```

7.1.16. Properties / WMO Data Policy

The `wmo:dataPolicy` property is a codelist that identifies the classification of the dataset exchange as described by WMO Unified Data Policy for the International Exchange of Earth System Data (Resolution 1 (Cg-Ext(2021) ^[33]). The codelist values are `core` or `recommended`. `wmo:dataPolicy` is required if the record describes a dataset.

Requirement 14	/req/core/data_policy
A	If a WCMP record's type (<code>properties.type</code>) is a dataset (i.e. <code>dataset</code>), the record SHALL provide exactly one <code>properties.wmo:dataPolicy</code> property.
B	A WCMP record's <code>properties.wmo:dataPolicy</code> property SHALL have a value of either <code>core</code> or <code>recommended</code> .
C	If a WCMP record's <code>properties.wmo:dataPolicy</code> is equal to <code>recommended</code> , any data licensing SHALL be provided by at least one <code>links</code> item with link relations (<code>rel</code>) of <code>license</code> .
D	If a WCMP record's <code>properties.wmo:dataPolicy</code> is equal to <code>recommended</code> , any copyright SHALL be provided by at least one <code>links</code> item with link relations (<code>rel</code>) of <code>copyright</code> .

Permission 6	/per/core/data_policy
A	A WCMP record whose data policy is <code>core</code> MAY provide attribution licensing as needed.

Licensing and copyright are expressed via the `links` property (see distribution section), providing access, license and attribution details as required. Conditions on use are essential for the WMO Unified Data Policy, but for transparency and clarification any other conditions should also be indicated.

Example: Core Data

```
"properties": {  
  ...  
  "wmo:dataPolicy": "core"  
  ...  
}
```

Example: Recommended Data

```
"properties": {  
  ...  
  "wmo:dataPolicy": "recommended"  
  ...  
}
```

Example: Recommended Data with additional conditions

```
"properties": {  
  ...  
  "wmo:dataPolicy": "recommended"  
  ...  
},  
"links": [{  
  "rel": "license",  
  "href": "https://example.org/dataset/license.html",  
  "type": "text/html",  
  "title": "Dataset license information"  
}]
```

It is useful to add provider-specific details to have the most detailed information about data policy and additional conditions.

Recommendation 10	/rec/core/data_policy_conditions
A	A WCMP record's additional conditions represented by a links item SHOULD provide a title property to provide additional human-readable information about link.

Example: Recommended Data with additional conditions and provider-specific details:

```
"properties": {  
  ...  
  "wmo:dataPolicy": "recommended"  
  ...  
},  
"links": [{  
  "rel": "license",  
  "href": "https://www.eumetsat.int/eumetsat-data-licensing",  
  "type": "text/html",  
  "title": "EUMETSAT DATA LICENSING"  
}]
```

7.1.17. Distribution Information

7.1.17.1. Overview

The **links** property describes URLs and APIs for accessing or subscribing to the dataset.

WCMP record links may also provide links to services that implement access control in support of authentication and authorization. In secure data use cases, a user needs to be able to detect access-controlled data as part of data discovery and evaluation.

Requirement 15	/req/core/links
A	A WCMP record SHALL define a links property.
B	A WCMP record links property SHALL contain at least one link to the data access service allowing users to download the data in one of the supported formats.
C	A WCMP record links property SHALL contain the MQTT topic information for real-time data under which the data publication notifications will be accessible from the WIS2 Global Broker, following the WIS2 topic hierarchy.
D	A WCMP record links property SHALL contain a Web Accessible Folder (WAF) OR an API link for non real-time data (e.g., climate records, hydrometric data archives).
E	A WCMP record links property SHALL contain access control information for data, products, and services requiring authentication or authorization.

F	For representing MQTT links, a WCMP record SHALL use the URI scheme of <code>mqtt</code> (default port 1883) and <code>mqtt</code> s (default port 8883) for secure TLS encrypted connections.
---	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Recommendation 11	/rec/core/links
A	A WCMP record's links SHOULD be provided using secure protocols (e.g. HTTPS, MQTTS, etc.).

Example

```
"links": [{
  "rel": "download",
  "type": "application/json",
  "title": "link to WAF endpoint",
  "href": "https://example.org/data/secure-data",
  "security": {
    "type": "http",
    "scheme": "basic",
    "description": "Please contact the data provider for accessing this secured resource."
  }
}]
```

Example: two link objects providing both search and MQTT subscription information

```
"links": [
  {
    "rel": "search",
    "type": "text/html",
    "title": "WOUDC - Data - Station List",
    "href": "https://woudc.org/data/stations"
  },
  {
    "rel" : "items",
    "type" : "application/json",
    "title": "WIS2 notification service",
    "href" : "mqtt://example.org",
    "channel": "cache/a/wis2/can/eccc-msc/data/core/weather/surface-based-observations"
  }
]
```

Permission 7	/per/core/links
---------------------	------------------------

A	A WCMP record links property MAY additionally contain a Web Accessible Folder (WAF) OR an API link for real-time data.
---	------------------------------------------------------------------------------------------------------------------------

Example: a link object providing an API capability

```
"links": [
  {
    "rel": "http://www.opengis.net/def/interface/ogcapi-features",
    "type": "application/json",
    "title": "OGC API - Features service",
    "href": "https://example.org/api"
  }
]
```

The **distribution** property provides information about the format of the dataset, associated documentation and representative examples of the dataset.

Recommendation 12	/rec/core/distribution
A	A WCMP record SHOULD describe additional distribution information with the properties.distribution property to qualify a given service link. This information SHOULD be used to build the discovery information and allow the user to find and choose the appropriate service for accessing the data.
B	The properties.distribution.availableFormat property SHOULD describe the different formats that can be retrieved using the service link.
C	The properties.distribution.availableFormat.description SHOULD describe the associated format.
D	The properties.distribution.availableFormat.typicalFilename SHOULD describe the filenames that are available from the service link.
E	The properties.distribution.availableFormat.typicalFileSize SHOULD describe the typical filesize available from the service link.
F	The properties.distribution.availableFormat.numberOfFile SHOULD describe the typical number of files received during a given period (day, month, ...) when using the service link. It SHOULD follow the pattern X per day, X per month.

G	The <code>properties.distribution.availableFormat.documentation</code> SHOULD be a link for accessing the documentation associated with the format.
H	The <code>properties.distribution.availableFormat.productSamples</code> SHOULD be an array of direct links to representative samples of the data.

Example: a link object of a WMS service providing API access to images:

```
"links": [{
  "rel": "item",
  "type": "image/png",
  "title": "Eumetview",
  "href":
  "https://view.eumetsat.int/geoserver/ows?service=WMS&request=GetMap&version=1.3.0&layers=msg_fes:ir108&styles=&format={format}&crs={crs}&bbox={bbox}&width={width}&height={height}",
  "templated": true,
  "variables": {
    "crs": {
      "description": "...",
      "type": "string",
      "enum": [
        "EPSG:4326",
        "EPSG:3857"
      ]
    },
    "bbox": {
      "description": "...",
      "type": "array",
      "items": {
        "type": "number",
        "format": "double"
      },
      "minItems": 4,
      "maxItems": 4
    },
    "width": {
      "description": "...",
      "type": "number",
      "format": "integer",
      "minimum": 600,
      "maximum": 5000
    },
    "height": {
      "description": "...",
      "type": "number",
      "format": "integer",

```



```

    "minimum": 600,
    "maximum": 5000
  },
  "format": {
    "description": "...",
    "type": "string",
    "enum": [
      "image/geotiff",
      "image/geotiff8",
      "image/gif",
      "image/jpeg",
      "image/png; mode=8bit"
    ]
  },
  "sampleRequest":
    "https://view.eumetsat.int/geoserver/ows?service=WMS&request=GetMap&version=1.3.0&layers=msg_fes:ir108&styles=&format=image/jpeg&crs=EPSG:4326&bbox=-77,-77,77,77&width=800&height=800"
  }
}]

```

Additional distribution information is added to allow more comprehensive discovery services. Below is an example of such distribution information.

Example with the complete distribution information

```
"links": [{
  "rel": "service",
  "type": "text/html",
  "title": "EUMETSAT Datastore",
  "href": "https://data.eumetsat.int/data/map/E0:EUM:DAT:MSG:MSG15-RSS",
  "distribution": {
    "availableFormats": [{
      "name": "native",
      "description": "This is sent in a compressed Submission Information Package (SIP) by default.",
      "numberOfFiles": "288 per day",
      "typicalFileSize": "60 MB",
      "typicalFilename": "MSG3-SEVI-MSG15-0100-NA-20130208102743.243000000Z-1051616.zip",
      "samples": "https://data.eumetsat.int/data/access/MSG3-SEVI-MSG15-0100-NA-20130208102743.243000000Z-1051616.zip",
      "documentation": {
        "rel": "alternate",
        "type": "text/html",
        "title": "SIP documentation and tools",
        "href": "https://www.eumetsat.int/formats#SIP"
      }
    }]
  }
}]
```

WCMP record links may also provide links to services that implement access control in support of authentication and authorization. In secure data use cases, a user needs to be able to detect access controlled data as part of data discovery and evaluation. The example demonstrates how to express access control using HTTP Basic Authentication for a given data access service.

Example

```
"links": [{
  "rel": "download",
  "type": "application/json",
  "title": "link to WAF endpoint",
  "href": "https://example.org/data/secure-data",
  "security": {
    "type": "http",
    "scheme": "basic",
    "description": "Please contact the data provider for accessing this secured resource."
  }
}]
```

7.1.18. Additional Properties

A WCMP record can be extended as required for organizational purposes by adding properties (of any type) in the record. Additional properties do not break compliance to WCMP.

Permission 8	/per/core/additional_properties
A	A WCMP record MAY provide additional properties of any type in any part of the document as needed.

Example

```
"properties": {  
  ...  
  "approvalStatus": "approved"  
  "_comment": {  
    "validationErrors": [  
      "error 1",  
      "error 2"  
    ]  
  }  
  ...  
}
```

[28] https://library.wmo.int/doc_num.php?explnum_id=11113#page=9

[29] <https://github.com/wmo-im/wcmp2-codelists/blob/main/codelists/resource-type.csv>

[30] <https://doi.org>

[31] <https://arks.org>

[32] <https://handle.net>

[33] https://library.wmo.int/doc_num.php?explnum_id=11113#page=9

Annex A: Conformance Class Abstract Test Suite (Normative)

A.1. Conformance Class: Core

label

<http://www.wmo.int/spec/wcmp/2/conf/core>

subject

Requirements Class "core"

classification

Target Type:Discovery Metadata

A.1.1. Validation

label

/conf/core/validation

subject

/req/core/validation

test-purpose

Validate that a WCMP record is valid to the authoritative WCMP schema.

Run JSON Schema validation on the WCMP record against the WCMP authoritative schema.

A.1.2. Identifier

label

/conf/core/identifier

subject

/req/core/identifier

test-purpose

Validate that a WCMP record has a valid identifier.

Check for the existence of an **id** property in the WCMP record.

In the WCMP record's **id** property, check that there are **five** tokens, delimited by **:**.

In the WCMP record's **id** property, delimiting the value on **:**, check that the first three tokens are equal to [**'urn'**, **'x-wmo'**, **'md'**].

In the WCMP record's **id** property, delimiting the value on **:**, check that the fourth token is a country based on the WIS2 Topic Hierarchy and controlled vocabulary.

In the WCMP record's **id** property, delimiting the value on **:**, check that the fifth token is a centre id based on the WIS2 Topic Hierarchy and controlled vocabulary.

In the WCMP record's **id** property, delimiting the value on **:**, check that the fifth token (the local identifier) has no spaces or accented characters.

A.1.3. Conformance

label

/conf/core/conformance

subject

/req/core/conformance

test-purpose

Validate that a WCMP record provides valid conformance information.

Check for the existence of a **conformsTo** property in the WCMP record.

In the WCMP record's **conformsTo** array property, check that ONE of the values is equal to <http://wmo.int/spec/wcmp/2/conf/core>.

A.1.4. Type

label

/conf/core/type

subject

/req/core/type

test-purpose

Validate that a WCMP record provides valid resource type information.

Check for the existence of a **properties.type** property in the WCMP record.

Check that the value of **properties.type** is part of the WCMP resource type codelist.

A.1.5. Geospatial Extent

label

/conf/core/extent_geospatial

subject

/req/core/extent_geospatial

test-purpose

Validate that a WCMP record provides a valid geometry property.

Check for the existence of one **geometry** property in the WCMP record.

Check that all **geometry** coordinate value data types are integers or floats.

Check that **geometry** coordinate longitudinal values are between -180 and 180.

Check that **geometry** coordinate latitudinal values are between -90 and 90.

Check that **geometry** property is a valid GeoJSON geometry.

A.1.6. Temporal Extent

label

/conf/core/extent_temporal

subject

/req/core/extent_temporal

test-purpose

Validate that a WCMP record provides a valid temporal extent property.

Check for the existence of one **time** item property in the WCMP record.

Check that the **time** object is one of **date** string, **timestamp** string, **interval** array, or **null**.

Check that all non-null **time** values are valid ISO8601 representations, intervals or durations, or **..** for an open-ended extent.

A.1.7. Title

label

/conf/core/title

subject

/req/core/title

test-purpose

Validate that a WCMP record provides a title property.

Check for the existence of a **properties.title** property in the WCMP record.

A.1.8. Description

label

/conf/core/description

subject

/req/core/description

test-purpose

Validate that a WCMP record provides a description property.

Check for the existence of a `properties.description` property in the WCMP record.

A.1.9. Themes

label

/conf/core/themes

subject

/req/core/themes

test-purpose

Validate that a WCMP record provides a themes property.

Check for the existence of a single `properties.themes` array property in the WCMP record.

Check that the `properties.themes` property provides a minimum of one theme object.

Check that each theme object provides a minimum of one `concepts` property.

Check that each theme object provides one `scheme` property.

Check that each concept object provides a minimum of one `id` property.

Check that the `properties.themes` property provides a theme object with the `scheme` of <https://github.com/wmo-im/wcmp2-codelists/blob/main/codelists/earth-system-discipline.csv>.

Check that all concepts identified are part of the `scheme` provided.

label

/conf/core/themes_wis2_global_service

subject

/req/core/themes_wis2_global_service

test-purpose

Validate that a WCMP record provides a themes property from a WIS2 global service.

Check for the existence of a **properties.type** element with a value of **service**.

Check that the **properties.themes** property provides a theme object with the **scheme** of <https://github.com/wmo-im/wcmp2-codelists/blob/main/codelists/earth-system-discipline.csv>.

Check that each value from the **scheme** is provided as a concept.

Check that the **properties.themes** property provides a theme object with the **scheme** of <https://github.com/wmo-im/wcmp2-codelists/blob/main/codelists/service-type.csv>.

Check that one concept from the **scheme** is provided.

A.1.10. Contacts

label

/conf/core/contacts

subject

/req/core/contacts

test-purpose

Validate that a WCMP record provides contact information for the metadata point of contact and originator of the data.

Check for the existence of a **properties.contacts** property in the WCMP record.

Check that the **contact** property provides a minimum of one contact object.

Check for the existence of a **roles** property in the contact object.

Check for the existence of an **organization** property in the contact object.

If **roles** exists, check that the value is part of the WCMP contact role codelist.

A.1.11. Record Creation Date

label

/conf/core/record_creation_date

subject

/req/core/record_creation_date

test-purpose

Validate that a WCMP record provides a record creation date.

Check for the existence of one and only one **properties.created** property in the WCMP record.

A.1.12. WMO Data Policy

label

/conf/core/data_policy

subject

/req/core/data_policy

test-purpose

Validate that a WCMP record provides information about data policy and, if applicable additional information about licensing and/or copyright.

Check for the type of WCMP record (`properties.type`).

If `properties.type` is found, and `properties.type` is equal to `dataset` check that `properties.wmo:dataPolicy` exists.

Check for the existence of a `properties.wmo:dataPolicy` property in the WCMP record.

If `properties.wmo:dataPolicy` is found, check that `properties.wmo:dataPolicy` has a value equal to `core` or `recommended`.

If `properties.wmo:dataPolicy` is equal to `recommended`, check for the existence of at least one `links` item containing a `license` or `copyright` link relation (`rel`).

A.1.13. Links

label

/conf/core/links

subject

/req/core/links

test-purpose

Validate that a WCMP record provides a link property.

Check for the existence of a single **links** array property in the WCMP record.

Check that the **links** property provides a minimum of one link object.

For a link object describing real-time data, check that the **href** property starts with the **mqtt** or **mqttts** protocol, AND that **channel** is additionally defined.

For a link object describing archived (NOT real-time) data made available via API, check that the **rel** property is a recognized API service type as defined by IANA or OGC.

For a link object describing archived (NOT real-time) data made available via Web Accessible Folder, check that the **rel** property is a recognized API service type as defined by IANA or OGC.

For a link object with access control (defined by the **security** object), check that the **security** property contains a **description** property with instructions on how to obtain access.

Annex B: Schemas (Normative)

NOTE

The schema document will be published on <http://schemas.wmo.int/wcmp/2.0> once the standard has been approved.

B.1. WMO Core Metadata Profile Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://raw.githubusercontent.com/wmo-
im/wcmp2/main/schema/wcmpRecordGeoJSON.yaml",
  "title": "WCMP discovery metadata record definition",
  "description": "WCMP discovery metadata record definition",
  "required": [
    "id",
    "conformsTo",
    "type",
    "time",
    "geometry",
    "properties",
    "links"
  ],
  "properties": {
    "id": {
      "type": "string",
      "description": "A unique identifier of the catalogue record.",
      "format": "uri"
    },
    "conformsTo": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "type": {
      "type": "string",
      "enum": [
        "Feature"
      ]
    },
    "time": {
      "oneOf": [
        {
          "enum": [
            null
          ]
        },
        {

```

```

"type": "object",
"properties": {
  "date": {
    "type": "string",
    "pattern": "^\\d{4}-\\d{2}-\\d{2}$"
  },
  "timestamp": {
    "type": "string",
    "pattern": "^\\d{4}-\\d{2}-\\d{2}T\\d{2}:\\d{2}:\\d{2}(?:\\.\\d+)?Z$"
  },
  "interval": {
    "type": "array",
    "minItems": 2,
    "maxItems": 2,
    "items": {
      "oneOf": [
        {
          "type": "string",
          "pattern": "^\\d{4}-\\d{2}-\\d{2}$"
        },
        {
          "type": "string",
          "pattern": "^\\d{4}-\\d{2}$"
        },
        {
          "type": "string",
          "pattern": "^\\d{4}$"
        },
        {
          "type": "string",
          "pattern": "^\\d{4}-\\d{2}-\\d{2}T\\d{2}:\\d{2}:\\d{2}(?:\\.\\d+)?Z$"
        },
        {
          "type": "string",
          "pattern": "^T\\d{2}(:\\d{2})?(:\\d{2})?(?:\\.\\d+)?Z$"
        },
        {
          "type": "string",
          "enum": [
            ".."
          ]
        }
      ]
    }
  },
  "resolution": {
    "type": "string",
    "description": "Minimum time period resolvable in the dataset, as an ISO 8601 duration",
    "example": [

```

```

        "P1D"
    ]
}
}
}
],
},
"geometry": {
    "oneOf": [
        {
            "enum": [
                null
            ]
        },
        {
            "oneOf": [
                {
                    "type": "object",
                    "required": [
                        "type",
                        "coordinates"
                    ],
                    "properties": {
                        "type": {
                            "type": "string",
                            "enum": [
                                "Point"
                            ]
                        },
                        "coordinates": {
                            "type": "array",
                            "minItems": 2,
                            "items": {
                                "type": "number"
                            }
                        }
                    }
                }
            ]
        },
        {
            "type": "object",
            "required": [
                "type",
                "coordinates"
            ],
            "properties": {
                "type": {
                    "type": "string",
                    "enum": [
                        "MultiPoint"
                    ]
                }
            },

```

```

        "coordinates": {
            "type": "array",
            "items": {
                "type": "array",
                "minItems": 2,
                "items": {
                    "type": "number"
                }
            }
        }
    },
    {
        "type": "object",
        "required": [
            "type",
            "coordinates"
        ],
        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "LineString"
                ]
            },
            "coordinates": {
                "type": "array",
                "minItems": 2,
                "items": {
                    "type": "array",
                    "minItems": 2,
                    "items": {
                        "type": "number"
                    }
                }
            }
        }
    },
    {
        "type": "object",
        "required": [
            "type",
            "coordinates"
        ],
        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "MultiLineString"
                ]
            },

```



```

        "coordinates": {
          "type": "array",
          "items": {
            "type": "array",
            "minItems": 2,
            "items": {
              "type": "array",
              "minItems": 2,
              "items": {
                "type": "number"
              }
            }
          }
        }
      },
    },
  },
  {
    "type": "object",
    "required": [
      "type",
      "coordinates"
    ],
    "properties": {
      "type": {
        "type": "string",
        "enum": [
          "Polygon"
        ]
      },
      "coordinates": {
        "type": "array",
        "items": {
          "type": "array",
          "minItems": 4,
          "items": {
            "type": "array",
            "minItems": 2,
            "items": {
              "type": "number"
            }
          }
        }
      }
    }
  }
},
{
  "type": "object",
  "required": [
    "type",
    "coordinates"
  ],

```

```

    "properties": {
      "type": {
        "type": "string",
        "enum": [
          "MultiPolygon"
        ]
      },
      "coordinates": {
        "type": "array",
        "items": {
          "type": "array",
          "items": {
            "type": "array",
            "minItems": 4,
            "items": {
              "type": "array",
              "minItems": 2,
              "items": {
                "type": "number"
              }
            }
          }
        }
      }
    }
  },
  {
    "type": "object",
    "required": [
      "type",
      "geometries"
    ],
    "properties": {
      "type": {
        "type": "string",
        "enum": [
          "GeometryCollection"
        ]
      },
      "geometries": {
        "type": "array",
        "items": {
          "oneOf": [
            {
              "type": "object",
              "required": [
                "type",
                "coordinates"
              ],
              "properties": {
                "type": {

```

```

        "type": "string",
        "enum": [
            "Point"
        ]
    },
    "coordinates": {
        "type": "array",
        "minItems": 2,
        "items": {
            "type": "number"
        }
    }
},
{
    "type": "object",
    "required": [
        "type",
        "coordinates"
    ],
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "MultiPoint"
            ]
        },
        "coordinates": {
            "type": "array",
            "items": {
                "type": "array",
                "minItems": 2,
                "items": {
                    "type": "number"
                }
            }
        }
    }
},
{
    "type": "object",
    "required": [
        "type",
        "coordinates"
    ],
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "LineString"
            ]
        }
    }
}

```

```

    },
    "coordinates": {
      "type": "array",
      "minItems": 2,
      "items": {
        "type": "array",
        "minItems": 2,
        "items": {
          "type": "number"
        }
      }
    }
  }
},
{
  "type": "object",
  "required": [
    "type",
    "coordinates"
  ],
  "properties": {
    "type": {
      "type": "string",
      "enum": [
        "MultiLineString"
      ]
    },
    "coordinates": {
      "type": "array",
      "items": {
        "type": "array",
        "minItems": 2,
        "items": {
          "type": "array",
          "minItems": 2,
          "items": {
            "type": "number"
          }
        }
      }
    }
  }
}
},
{
  "type": "object",
  "required": [
    "type",
    "coordinates"
  ],
  "properties": {
    "type": {

```

```

        "type": "string",
        "enum": [
            "Polygon"
        ]
    },
    "coordinates": {
        "type": "array",
        "items": {
            "type": "array",
            "minItems": 4,
            "items": {
                "type": "array",
                "minItems": 2,
                "items": {
                    "type": "number"
                }
            }
        }
    }
},
{
    "type": "object",
    "required": [
        "type",
        "coordinates"
    ],
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "MultiPolygon"
            ]
        },
        "coordinates": {
            "type": "array",
            "items": {
                "type": "array",
                "items": {
                    "type": "array",
                    "minItems": 4,
                    "items": {
                        "type": "array",
                        "minItems": 2,
                        "items": {
                            "type": "number"
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
  },
  {
    "type": "object",
    "required": [
      "type",
      "geometries"
    ],
    "properties": {
      "type": {
        "type": "string",
        "enum": [
          "GeometryCollection"
        ]
      },
      "geometries": {
        "type": "array",
        "items": {
          "$ref":
            "#/properties/geometry/oneOf/1/oneOf/6/properties/geometries/items"
        }
      }
    }
  }
]
},
"additionalExtents": {
  "description": "The extent of the information in the collection. In the Core
only spatial and temporal\nextents are specified. Extensions may add additional
members to represent other\nextents, for example, thermal or pressure ranges.",
  "type": "object",
  "properties": {
    "spatial": {
      "description": "The spatial extent of the information in the collection.",
      "type": "object",
      "required": [
        "bbox",
        "crs"
      ],
      "properties": {
        "bbox": {
          "description": "One or more bounding boxes that describe the spatial
extent of the dataset.\nIn the Core only a single bounding box is supported.
Extensions may support\nadditional areas. If multiple areas are provided, the union of

```

the bounding\boxes describes the spatial extent.",

```
"type": "array",
```

```
"minItems": 1,
```

```
"items": {
```

```
  "description": "Each bounding box is provided as four or six numbers, depending on whether the coordinate reference system includes a vertical axis\n(height or depth):\n\n* Lower left corner, coordinate axis 1\n* Lower left corner, coordinate axis 2\n* Minimum value, coordinate axis 3 (optional)\n* Upper right corner, coordinate axis 1\n* Upper right corner, coordinate axis 2\n* Maximum value, coordinate axis 3 (optional)\n\nThe coordinate reference system of the values is WGS 84 longitude/latitude\n(http://www.opengis.net/def/crs/OGC/1.3/CRS84) unless a different coordinate\nreference system is specified in `crs`.\n\nFor WGS 84 longitude/latitude the values are in most cases the sequence of\nminimum longitude, minimum latitude, maximum longitude and maximum latitude.\nHowever, in cases where the box spans the antimeridian the first value\n(west-most box edge) is larger than the third value (east-most box edge).\n\nIf the vertical axis is included, the third and the sixth number are\nthe bottom and the top of the 3-dimensional bounding box.\n\nIf a feature has multiple spatial geometry properties, it is the decision of the\nserver whether only a single spatial geometry property is used to determine\nthe extent or all relevant geometries.",
```

```
  "oneOf": [
```

```
    {
```

```
      "items": {
```

```
        "type": "number"
```

```
      },
```

```
      "minItems": 4,
```

```
      "maxItems": 4,
```

```
      "type": "array"
```

```
    },
```

```
    {
```

```
      "items": {
```

```
        "type": "number"
```

```
      },
```

```
      "minItems": 6,
```

```
      "maxItems": 6,
```

```
      "type": "array"
```

```
    }
```

```
  ]
```

```
},
```

```
"example": [
```

```
  -180,
```

```
  -90,
```

```
  180,
```

```
  90
```

```
]
```

```
},
```

```
"crs": {
```

```
  "description": "Coordinate reference system of the coordinates in the spatial extent\n(property `bbox`). The default reference system is WGS 84 longitude/latitude.\nIn the Core this is the only supported coordinate reference system.\nExtensions may support additional coordinate reference systems ",
```

```

        "type": "string",
        "default": "GEOGCS[\"WGS 84\",DATUM[\"WGS_1984\",SPHEROID[\"WGS
84\",6378137,298.257223563,AUTHORITY[\"EPSG\", \"7030\"]],AUTHORITY[\"EPSG\", \"6326\"]],PRIMEM[\"Greenwich\",0,AUTHORITY[\"EPSG\", \"8901\"]],UNIT[\"degree\",0.01745329251994
328,AUTHORITY[\"EPSG\", \"9122\"]],AUTHORITY[\"EPSG\", \"4326\"]]"
    },
    "name": {
        "description": "Name of the vertical coordinate reference system",
        "type": "string"
    }
}
},
"temporal": {
    "description": "The temporal extent of the information in the collection.",
    "type": "object",
    "required": [
        "interval",
        "trs"
    ],
    "properties": {
        "interval": {
            "description": "RFC3339 compliant Date and Time",
            "type": "array",
            "minItems": 1,
            "items": {
                "description": "Begin and end times of the time interval. The
timestamps are in the\ntemporal coordinate reference system specified in `trs`. By
default\nthis is the Gregorian calendar.",
                "type": "array",
                "minItems": 2,
                "maxItems": 2,
                "items": {
                    "type": "string",
                    "format": "date-time",
                    "nullable": true
                }
            },
            "example": [
                [
                    "2020-11-12T12:15Z",
                    "2020-11-12T12:15Z"
                ],
                [
                    "2020-11-12T12:15Z",
                    "2020-11-12T12:45Z"
                ],
                [
                    "2022-01-14T09:00Z",
                    "2022-01-15T09:00Z"
                ],
                [
                    "2022-01-14T09:00Z",

```



```

        "2022-01-14T21:00Z"
      ],
      [
        "2022-01-14T09:00Z",
        "2022-01-16T09:00Z"
      ]
    ]
  },
  "values": {
    "description": "Provides information about the time intervals available
in the collection \nas ISO8601 compliant dates, either as a time range specified \nas
start time / end time (e.g. 2017-11-14T09:00Z/2017-11-14T21:00Z) or\nas number of
repetitions / start time / interval (e.g. R4/2017-11-14T21:00Z/PT3H) \nor a list of
time values (e.g.\n2017-11-14T09:00Z,2017-11-14T12:00Z,2017-11-14T15:00Z,2017-11-
14T18:00Z,2017-11-14T21:00Z)",
    "type": "array",
    "minItems": 1,
    "items": {
      "type": "string",
      "minItems": 1,
      "format": "date-time",
      "nullable": true,
      "example": [
        [
          "2020-11-12T12:15Z"
        ],
        [
          "2020-11-12T12:15Z",
          "2020-11-12T12:30Z",
          "2020-11-12T12:45Z"
        ],
        [
          "R12/2022-01-14T09:00Z/2022-01-15T09:00Z"
        ],
        [
          "R12/2022-01-14T09:00Z/PT1H"
        ],
        [
          "R12/2022-01-14T09:00Z/PT1H",
          "R4/2022-01-14T21:00Z/PT3H",
          "R4/2022-01-15T09:00Z/PT6H"
        ]
      ]
    }
  },
  "trs": {
    "description": "Coordinate reference system of the coordinates in the
temporal extent\n(property `interval`). The default reference system is the Gregorian
calendar.\nIn the Core this is the only supported temporal coordinate reference
system.\nExtensions may support additional temporal coordinate reference systems ",

```

```

        "type": "string",
        "example": "TIMECRS[\"DateTime\",TDATUM[\"Gregorian
Calendar\"],CS[TemporalDateTime,1],AXIS[\"Time (T)\",future]]",
        "default": "http://www.opengis.net/def/uom/ISO-8601/0/Gregorian"
    },
    "name": {
        "description": "Name of the temporal coordinate reference system",
        "type": "string"
    }
}
},
"vertical": {
    "description": "The vertical extent of the information in the collection.",
    "type": "object",
    "required": [
        "interval",
        "vrs"
    ],
    "properties": {
        "interval": {
            "description": "In the Core only a single time interval is supported.
Extensions may support\nmultiple intervals. If multiple intervals are provided, the
union of the\nintervals describes the vertical extent.",
            "type": "array",
            "minItems": 1,
            "items": {
                "description": "minimum and maximum heights of the vertical interval.
The values are in the\nvertical coordinate reference system specified in `vrs`. By
default\nthis is the vertical coordinate reference system of the source information.",
                "type": "array",
                "minItems": 1,
                "items": {
                    "type": "string",
                    "nullable": true
                }
            },
            "example": [
                [
                    "2",
                    "100"
                ]
            ]
        }
    },
    "values": {
        "description": "Vertical level intervals that data in the collection is
available at \nthese can be defined as follows:\nmin level / max level (e.g.
\"2/100\") or \nas number of repetitions / start level / interval (e.g.\"R5/100/50\")
\nor a list of vertical levels (e.g. \"2\",10,\"80\", \"100\")\nThe value `null` is
supported and indicates an open vertical interval.",
        "type": "array",
        "minItems": 1,

```

```

        "items": {
            "type": "string",
            "nullable": true,
            "example": [
                [
                    "2/100"
                ],
                [
                    "R20/1000/-50"
                ],
                [
                    "2",
                    "10",
                    "80",
                    "100"
                ]
            ]
        },
        "vrs": {
            "description": "Coordinate reference system of the coordinates in the vertical extent\n(property `interval`). ",
            "type": "string",
            "example": "VERTCS[\"WGS_1984\",DATUM[\"D_WGS_1984\",SPHEROID[\"WGS_1984\",6378137.0,298.257223563]],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"Meter\",1.0],AXIS[\"Up\",UP]\",
            "default": "VERTCS[\"WGS_1984\",DATUM[\"D_WGS_1984\",SPHEROID[\"WGS_1984\",6378137.0,298.257223563]],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"Meter\",1.0],AXIS[\"Up\",UP]\"
        },
        "name": {
            "description": "Name of the vertical coordinate reference system",
            "type": "string"
        }
    }
},
"properties": {
    "type": "object",
    "required": [
        "type",
        "title",
        "description",
        "contacts",
        "created"
    ],
    "properties": {
        "type": {

```

```

    "type": "string",
    "description": "The nature or genre of the resource. The value should be a
code, convenient for filtering records. Where available, a link to the canonical URI
of the record type resource will be added to the 'links' property.",
    "maxLength": 64
  },
  "title": {
    "type": "string",
    "description": "A human-readable name given to the resource."
  },
  "description": {
    "type": "string",
    "description": "A free-text account of the resource."
  },
  "keywords": {
    "type": "array",
    "description": "The topic or topics of the resource. Typically represented
using free-form keywords, tags, key phrases, or classification codes.",
    "items": {
      "type": "string"
    }
  },
  "themes": {
    "type": "array",
    "minItems": 1,
    "items": {
      "type": "object",
      "required": [
        "concepts",
        "scheme"
      ],
      "properties": {
        "concepts": {
          "type": "array",
          "description": "One or more entity/concept identifiers from this
knowledge system. it is recommended that a resolvable URI be used for each
entity/concept identifier.",
          "minItems": 1,
          "items": {
            "type": "object",
            "required": [
              "id"
            ],
            "properties": {
              "id": {
                "type": "string",
                "description": "An identifier for the concept."
              },
              "title": {
                "type": "string",
                "description": "A human readable title for the concept."
              }
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "description": {
      "type": "string",
      "description": "A human readable description for the concept."
    },
    "url": {
      "type": "string",
      "format": "uri",
      "description": "A URI providing further description of the
concept."
    }
  }
},
"scheme": {
  "type": "string",
  "description": "An identifier for the knowledge organization system
used to classify the resource. It is recommended that the identifier be a resolvable
URI. The list of schemes used in a searchable catalogue can be determined by
inspecting the server's OpenAPI document or, if the server implements CQL2, by
exposing a queryable (e.g. named `scheme`) and enumerating the list of schemes in the
queryable's schema definition."
}
},
"contacts": {
  "type": "array",
  "minItems": 1,
  "items": {
    "allOf": [
      {
        "required": [
          "organization"
        ],
        "type": "object",
        "description": "Identification of, and means of communication with,
person responsible\nfor the resource.",
        "properties": {
          "identifier": {
            "type": "string",
            "description": "A value uniquely identifying a contact (individual
or organization)"
          },
          "name": {
            "type": "string",
            "description": "The name of the organization or the individual."
          },
          "positionName": {
            "type": "string",
            "description": "The name of the role or position of the

```

responsible person taken from the organization's formal organizational hierarchy or chart."

```
    },
    "organization": {
      "type": "string",
      "description": "Organization/affiliation of the
individual/responsible person. In case of an organization, the name property should be
used and this property is not to be used."
    },
    "logo": {
      "description": "Graphic identifying a contact. The link relation
should be `icon` and the media type should be an image media type.",
      "allOf": [
        {
          "type": "object",
          "required": [
            "href"
          ],
          "properties": {
            "href": {
              "type": "string",
              "format": "url"
            },
            "rel": {
              "type": "string"
            },
            "type": {
              "type": "string"
            },
            "hreflang": {
              "type": "string"
            },
            "title": {
              "type": "string"
            },
            "templated": {
              "type": "boolean"
            },
            "variables": {
              "description": "This object contains one key per
substitution variable in a templated\nhref. Each key defines the schema of one
substitution variable using\na JSON Schema fragment and can thus include things like
the data type\nof the variable, enumerations, minimum values, maximum values, etc.\nIn
combination with a templated href, the variables section should\nprovide enough
information to bind to the target resource (e.g. a WMS).",
              "type": "object"
            },
            "created": {
              "type": "string",
              "description": "Date of creation of the resource pointed
to by the link.",
            }
          }
        }
      ]
    }
  }
}
```

```

        "format": "date-time"
    },
    "updated": {
        "type": "string",
        "description": "Most recent date on which the resource
pointed to by the link was changed.",
        "format": "date-time"
    }
},
{
    "type": "object",
    "required": [
        "rel",
        "type"
    ],
    "properties": {
        "rel": {
            "enum": [
                "icon"
            ]
        }
    }
}
],
},
"phones": {
    "type": "array",
    "description": "Telephone numbers at which contact can be made.",
    "items": {
        "type": "object",
        "required": [
            "value"
        ],
        "properties": {
            "value": {
                "type": "string",
                "description": "The value is the phone number itself.",
                "pattern": "^\\+[1-9]{1}[0-9]{3,14}$",
                "example": "+14165550142"
            },
            "roles": {
                "description": "The type of phone number (e.g. home, work,
fax, etc.).",
                "type": "array",
                "items": {
                    "type": "string"
                }
            }
        }
    }
}
}

```

```

    },
    "emails": {
      "type": "array",
      "description": "Email addresses at which contact can be made.",
      "items": {
        "type": "object",
        "required": [
          "value"
        ],
        "properties": {
          "value": {
            "type": "string",
            "description": "The value is the email number itself.",
            "format": "email"
          },
          "roles": {
            "description": "The type of email (e.g. home, work, etc.).",
            "type": "array",
            "items": {
              "type": "string"
            }
          }
        }
      }
    },
    "addresses": {
      "type": "array",
      "description": "Physical location at which contact can be made.",
      "items": {
        "type": "object",
        "properties": {
          "deliveryPoint": {
            "type": "array",
            "description": "Address lines for the location.",
            "items": {
              "type": "string"
            }
          },
          "city": {
            "type": "string",
            "description": "City for the location."
          },
          "administrativeArea": {
            "type": "string",
            "description": "State or province of the location."
          },
          "postalCode": {
            "type": "string",
            "description": "ZIP or other postal code."
          },
          "country": {

```



```

        "type": "string",
        "description": "Country of the physical address."
    },
    "roles": {
        "description": "The type of address (e.g. office, home,
etc.).",
        "type": "array",
        "items": {
            "type": "string"
        }
    }
}

```

```

},
"links": {
    "type": "array",
    "description": "On-line information about the contact.",
    "items": {
        "allOf": [
            {
                "type": "object",
                "required": [
                    "href"
                ],
                "properties": {
                    "href": {
                        "type": "string",
                        "format": "url"
                    },
                    "rel": {
                        "type": "string"
                    },
                    "type": {
                        "type": "string"
                    },
                    "hreflang": {
                        "type": "string"
                    },
                    "title": {
                        "type": "string"
                    },
                    "templated": {
                        "type": "boolean"
                    }
                },
                "variables": {

```

"description": "This object contains one key per substitution variable in a templated\href. Each key defines the schema of one substitution variable using\ na JSON Schema fragment and can thus include things like the data type\ nof the variable, enumerations, minimum values, maximum values, etc.\ nIn combination with a templated href, the variables section should\ nprovide enough information to bind to the target resource (e.g. a WMS).",

```

        "type": "object"
      },
      "created": {
        "type": "string",
        "description": "Date of creation of the resource pointed
to by the link.",
        "format": "date-time"
      },
      "updated": {
        "type": "string",
        "description": "Most recent date on which the resource
pointed to by the link was changed.",
        "format": "date-time"
      }
    },
    {
      "type": "object",
      "required": [
        "type"
      ]
    }
  ]
},
"hoursOfService": {
  "type": "string",
  "description": "Time period when the contact can be contacted.",
  "example": "Hours: Mo-Fr 10am-7pm Sa 10am-22pm Su 10am-21pm"
},
"contactInstructions": {
  "type": "string",
  "description": "Supplemental instructions on how or when to
contact the\nresponsible party."
},
"roles": {
  "description": "The set of named duties, job functions and/or
permissions associated with this contact. (e.g. developer, administrator, etc.).",
  "type": "array",
  "items": {
    "type": "string"
  }
}
}
]
}
},
"version": {
  "type": "string",
  "description": "The version or edition of a given dataset."
}
}

```

```

    },
    "externalIds": {
      "type": "array",
      "description": "An identifier for the resource assigned by an external (to
the catalogue) entity.",
      "items": {
        "type": "object",
        "properties": {
          "scheme": {
            "type": "string",
            "description": "A reference to an authority or identifier for a
knowledge organization system from which the external identifier was obtained. It is
recommended that the identifier be a resolvable URI."
          },
          "value": {
            "type": "string",
            "description": "The value of the identifier."
          }
        }
      },
      "required": [
        "value"
      ]
    },
    "created": {
      "type": "string",
      "description": "Date of creation of this record.",
      "format": "date-time"
    },
    "updated": {
      "type": "string",
      "description": "The most recent date on which the record was changed.",
      "format": "date-time"
    },
    "wmo:dataPolicy": {
      "type": "string",
      "description": "The data policy definition as per the WMO Unified Data
Policy Resolution (Res.1) [23].",
      "enum": [
        "core",
        "recommended"
      ]
    }
  },
  "links": {
    "type": "array",
    "minItems": 1,
    "items": {
      "$schema": "https://json-schema.org/draft/2020-12/schema",
      "$id": "https://raw.githubusercontent.com/wmo-im/wcmp2/main/schema/link.yaml",

```

```

"title": "WCMP link object definition",
"description": "WCMP link object definition",
"required": [
  "href"
],
"properties": {
  "href": {
    "type": "string",
    "format": "url"
  },
  "rel": {
    "type": "string"
  },
  "type": {
    "type": "string"
  },
  "hreflang": {
    "type": "string"
  },
  "title": {
    "type": "string"
  },
  "templated": {
    "type": "boolean"
  },
  "variables": {
    "description": "This object contains one key per substitution variable in
a templated\nhref. Each key defines the schema of one substitution variable using\nna
JSON Schema fragment and can thus include things like the data type\nof the variable,
enumerations, minimum values, maximum values, etc.\nIn combination with a templated
href, the variables section should\nprovide enough information to bind to the target
resource (e.g. a WMS).",
    "type": "object"
  },
  "created": {
    "type": "string",
    "description": "Date of creation of the resource pointed to by the link.",
    "format": "date-time"
  },
  "updated": {
    "type": "string",
    "description": "Most recent date on which the resource pointed to by the
link was changed.",
    "format": "date-time"
  },
  "channel": {
    "type": "string",
    "description": "topic to subscribe to for broker workflow"
  },
  "security": {
    "type": "object",

```

```

"patternProperties": {
  "^[a-zA-Z0-9\\.\\-\\_]+$": {
    "oneOf": [
      {
        "type": "object",
        "required": [
          "$ref"
        ],
        "patternProperties": {
          "\\$ref$": {
            "type": "string",
            "format": "uri-reference"
          }
        }
      },
      {
        "oneOf": [
          {
            "type": "object",
            "required": [
              "type",
              "name",
              "in"
            ],
            "properties": {
              "type": {
                "type": "string",
                "enum": [
                  "apiKey"
                ]
              },
              "name": {
                "type": "string"
              },
              "in": {
                "type": "string",
                "enum": [
                  "header",
                  "query",
                  "cookie"
                ]
              },
              "description": {
                "type": "string"
              }
            },
            "patternProperties": {
              "^x-": {}
            },
            "additionalProperties": false
          },

```

```

{
  "type": "object",
  "required": [
    "scheme",
    "type"
  ],
  "properties": {
    "scheme": {
      "type": "string"
    },
    "bearerFormat": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "type": {
      "type": "string",
      "enum": [
        "http"
      ]
    }
  },
  "patternProperties": {
    "^X-": {}
  },
  "additionalProperties": false,
  "oneOf": [
    {
      "description": "Bearer",
      "properties": {
        "scheme": {
          "enum": [
            "bearer"
          ]
        }
      }
    },
    {
      "description": "Non Bearer",
      "not": {
        "required": [
          "bearerFormat"
        ]
      },
      "properties": {
        "scheme": {
          "not": {
            "enum": [
              "bearer"
            ]
          }
        }
      }
    }
  ]
}

```

```

    }
  }
}
]
},
{
  "type": "object",
  "required": [
    "type",
    "flows"
  ],
  "properties": {
    "type": {
      "type": "string",
      "enum": [
        "oauth2"
      ]
    },
    "flows": {
      "type": "object",
      "properties": {
        "implicit": {
          "type": "object",
          "required": [
            "authorizationUrl",
            "scopes"
          ],
          "properties": {
            "authorizationUrl": {
              "type": "string",
              "format": "uri-reference"
            },
            "refreshUrl": {
              "type": "string",
              "format": "uri-reference"
            },
            "scopes": {
              "type": "object",
              "additionalProperties": {
                "type": "string"
              }
            }
          },
          "patternProperties": {
            "^x-": {}
          },
          "additionalProperties": false
        },
        "password": {
          "type": "object",

```

```

    "required": [
      "tokenUrl"
    ],
    "properties": {
      "tokenUrl": {
        "type": "string",
        "format": "uri-reference"
      },
      "refreshUrl": {
        "type": "string",
        "format": "uri-reference"
      },
      "scopes": {
        "type": "object",
        "additionalProperties": {
          "type": "string"
        }
      }
    },
    "patternProperties": {
      "^x-": {}
    },
    "additionalProperties": false
  },
  "clientCredentials": {
    "type": "object",
    "required": [
      "tokenUrl"
    ],
    "properties": {
      "tokenUrl": {
        "type": "string",
        "format": "uri-reference"
      },
      "refreshUrl": {
        "type": "string",
        "format": "uri-reference"
      },
      "scopes": {
        "type": "object",
        "additionalProperties": {
          "type": "string"
        }
      }
    },
    "patternProperties": {
      "^x-": {}
    },
    "additionalProperties": false
  },
  "authorizationCode": {

```



```

        "type": "object",
        "required": [
            "authorizationUrl",
            "tokenUrl"
        ],
        "properties": {
            "authorizationUrl": {
                "type": "string",
                "format": "uri-reference"
            },
            "tokenUrl": {
                "type": "string",
                "format": "uri-reference"
            },
            "refreshUrl": {
                "type": "string",
                "format": "uri-reference"
            },
            "scopes": {
                "type": "object",
                "additionalProperties": {
                    "type": "string"
                }
            }
        },
        "patternProperties": {
            "^x-": {}
        },
        "additionalProperties": false
    },
    "patternProperties": {
        "^x-": {}
    },
    "additionalProperties": false
},
"description": {
    "type": "string"
}
},
"patternProperties": {
    "^x-": {}
},
"additionalProperties": false
},
{
    "type": "object",
    "required": [
        "type",
        "openIdConnectUrl"
    ],

```

```

        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "openIdConnect"
                ]
            },
            "openIdConnectUrl": {
                "type": "string",
                "format": "uri-reference"
            },
            "description": {
                "type": "string"
            }
        },
        "patternProperties": {
            "^x-": {}
        },
        "additionalProperties": false
    }
}
}
}
},
"distribution": {
    "type": "object",
    "description": "the additional information qualifying the service and
allowing to build and display the \"access\" information in a portal",
    "properties": {
        "availableFormats": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "name": {
                        "type": "string",
                        "description": "available format short name."
                    },
                    "description": {
                        "type": "string",
                        "description": "the individual format description to allow
building the portal information"
                    }
                },
                "numberOfFiles": {
                    "type": "string",
                    "description": "typical number of files disseminated by the
distribution mechanism over a period of time"
                },
                "typicalFileSize": {

```

```

    "type": "string",
    "description": "size of a typical individual file (e.g. KB, MB,
GB)"
  },
  "typicalFilename": {
    "type": "string",
    "description": "example filename"
  },
  "samples": {
    "type": "object",
    "description": "links of samples that can be openly accessed by
users to provide a better understanding of the data",
    "items": {
      "type": "object",
      "required": [
        "href"
      ],
      "properties": {
        "href": {
          "type": "string",
          "format": "url"
        },
        "rel": {
          "type": "string"
        },
        "type": {
          "type": "string"
        },
        "hreflang": {
          "type": "string"
        },
        "title": {
          "type": "string"
        },
        "templated": {
          "type": "boolean"
        },
        "variables": {
          "description": "This object contains one key per
substitution variable in a templated\nhref. Each key defines the schema of one
substitution variable using\na JSON Schema fragment and can thus include things like
the data type\nof the variable, enumerations, minimum values, maximum values, etc.\nIn
combination with a templated href, the variables section should\nprovide enough
information to bind to the target resource (e.g. a WMS).",
          "type": "object"
        },
        "created": {
          "type": "string",
          "description": "Date of creation of the resource pointed
to by the link.",
          "format": "date-time"
        }
      }
    }
  }
}

```

```

    },
    "updated": {
      "type": "string",
      "description": "Most recent date on which the resource
pointed to by the link was changed.",
      "format": "date-time"
    }
  }
},
"documentation": {
  "type": "array",
  "description": "links to the associated documentation available
for this format when applicable",
  "items": {
    "type": "object",
    "required": [
      "href"
    ],
    "properties": {
      "href": {
        "type": "string",
        "format": "url"
      },
      "rel": {
        "type": "string"
      },
      "type": {
        "type": "string"
      },
      "hreflang": {
        "type": "string"
      },
      "title": {
        "type": "string"
      },
      "templated": {
        "type": "boolean"
      },
      "variables": {
        "description": "This object contains one key per
substitution variable in a templated\nhref. Each key defines the schema of one
substitution variable using\na JSON Schema fragment and can thus include things like
the data type\nof the variable, enumerations, minimum values, maximum values, etc.\nIn
combination with a templated href, the variables section should\nprovide enough
information to bind to the target resource (e.g. a WMS).",
        "type": "object"
      },
      "created": {
        "type": "string",
        "description": "Date of creation of the resource pointed

```

```
to by the link.",
    "format": "date-time"
  },
  "updated": {
    "type": "string",
    "description": "Most recent date on which the resource
pointed to by the link was changed.",
    "format": "date-time"
  }
}
}
```

Annex C: Examples (Informative)

C.1. WMO Core Metadata Profile Examples

Example: Canadian Hourly Surface Weather Observations (dataset)

```
{
  "id": "urn:x-wmo:md:can:eccc-msc:weather.observations.swob-realtime",
  "conformsTo": [
    "http://wis.wmo.int/spec/wcmp/2/conf/core"
  ],
  "time": {
    "interval": [
      "2010-11-11T11:11:11Z",
      ".."
    ]
  },
  "type": "Feature",
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          -142,
          28
        ],
        [
          -142,
          82
        ],
        [
          -52,
          82
        ],
        [
          -52,
          28
        ],
        [
          -142,
          28
        ]
      ]
    ]
  },
  "properties": {
    "title": "Surface Weather Observations",
    "description": "Surface Observations measured at the automatic and manual stations of the Environment and Climate Change Canada and partners networks, either
```

for a single station, or for the stations of specific provinces and territories (last 30 days)",

```
    "themes": [
      {
        "concepts": [
          {
            "id": "Weather"
          },
          {
            "id": "Archives"
          },
          {
            "id": "Precipitation"
          },
          {
            "id": "Air temperature"
          },
          {
            "id": "Humidity"
          },
          {
            "id": "Snow"
          },
          {
            "id": "Wind"
          },
          {
            "id": "Meteorological data"
          }
        ],
        "scheme": "https://canada.multites.net/cst"
      },
      {
        "concepts": [
          {
            "id": "weather"
          }
        ],
        "scheme": "https://github.com/wmo-im/wcmp2-codelists/blob/main/codelists/earth-system-domain.csv"
      }
    ],
    "contacts": [
      {
        "name": "National Inquiry Response Team",
        "organization": "Government of Canada; Environment and Climate Change Canada; Meteorological Service of Canada",
        "phones": [
          {
            "value": "+18199972800"
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "emails": [
      {
        "value": "enviroinfo@ec.gc.ca"
      }
    ],
    "addresses": [
      {
        "deliveryPoint": [
          "77 Westmorland Street, suite 260"
        ],
        "city": "Fredericton",
        "administrativeArea": "NB",
        "postalCode": "E3B 6Z4",
        "country": "Canada"
      }
    ],
    "links": [
      {
        "rel": "canonical",
        "type": "text/html",
        "href": "https://www.canada.ca/en/environment-climate-
change.html"
      }
    ],
    "roles": [
      "producer"
    ]
  }
],
"type": "dataset",
"created": "2018-01-01",
"updated": "2022-06-22",
"wmo:dataPolicy": "core"
},
"links": [
  {
    "rel": "http://www.wmo.int/def/rel/wmdr/1.0/FacilitySet",
    "href": "https://dd.weather.gc.ca/observations/doc/swob-
xml_station_list.csv",
    "type": "text/csv",
    "title": "Stations associated with this dataset"
  },
  {
    "rel": "download",
    "href": "https://dd.weather.gc.ca/observations/swob-ml",
    "type": "text/html",
    "hreflang": "en",
    "title": "Raw data download (CSV files)"
  },
  {

```



```

        "rel": "items",
        "href": "https://api.weather.gc.ca/collections/swob-realtime/items",
        "type": "application/json",
        "title": "Data access API interface"
    },
    {
        "rel": "related",
        "href": "https://eccc-msc.github.io/open-data/msc-
data/obs_station/readme_obs_insitu_swobdatamart_en",
        "type": "text/html",
        "title": "Documentation"
    },
    {
        "rel": "items",
        "href": "mqtt://example.org",
        "channel": "origin/a/wis2/can/eccc-
msc/data/core/weather/observations/surface-land/landFixed",
        "type": "application/json",
        "title": "Data notifications"
    }
]
}

```

Example: ICON-EPS GRIB data (dataset)

```

{
  "id": "urn:x-wmo:md:deu:dwd-offenbach:icon-eps.ALL",
  "type": "Feature",
  "conformsTo": [
    "http://wis.wmo.int/spec/wcmp/2/conf/core"
  ],
  "time": {
    "interval": [
      "T00Z",
      "T12Z"
    ],
    "resolution": "PT6H"
  },
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          -180,
          -90
        ],
        [
          -180,
          90
        ],
        [

```

```

        180,
        90
    ],
    [
        180,
        -90
    ],
    [
        -180,
        -90
    ]
]
],
},
"additionalExtents": {
    "spatial": {
        "bbox": [
            [
                -180,
                -90,
                180,
                90
            ]
        ],
        "resolution": "0.5*0.5",
        "crs": "http://www.opengis.net/def/crs/OGC/1.3/CRS84"
    },
    "temporal": {
        "interval": [
            [
                "R/T00Z",
                "PT48H"
            ],
            [
                "R/T12Z",
                "PT48H"
            ]
        ],
        "resolution": "PT1H",
        "trs": "http://www.opengis.net/def/trs/ISO-8601"
    }
},
"properties": {
    "type": "dataset",
    "title": "ICON-EPS GRIB data",
    "description": "ICON-EPS 0.5\u00b0 x 0.5\u00b0 regular lat/lon grid, up to +180h every 6h, runs 00/12 UTC, various parameter, various level, various threshold",
    "themes": [
        {
            "concepts": [
                {

```

```

        "id": "meteorology"
      }
    ],
    "scheme":
"http://wis.wmo.int/2012/codelists/WMOCodeLists#WMO_CategoryCode"
  },
  {
    "concepts": [
      {
        "id": "Meteorological geographical features"
      }
    ],
    "scheme": "https://inspire.ec.europa.eu/theme/mf"
  },
  {
    "concepts": [
      {
        "id": "FM 92 GRIB edition 2"
      }
    ],
    "scheme": "http://codes.wmo.int/codeform"
  },
  {
    "concepts": [
      {
        "id": "continual"
      }
    ],
    "scheme":
"http://standards.iso.org/iso/19139/resources/gmxCodelists.xml#MD_FrequencyCode"
  }
],
"contacts": [
  {
    "identifier": "DWD",
    "organization": "Deutscher Wetterdienst",
    "positionName": "Michael Denhard",
    "emails": [
      {
        "value": "wis@dwd.de"
      }
    ],
    "addresses": [
      {
        "deliveryPoint": [
          "Frankfurter Strasse 135"
        ],
        "city": "Offenbach",
        "postalCode": "63067",
        "country": "Germany"
      }
    ]
  }
]

```

```

    ],
    "links": [
      {
        "type": "text/html",
        "href": "https://www.dwd.de/wmc"
      },
      {
        "type": "text/html",
        "href": "https://gisc.dwd.de"
      }
    ],
    "contactInstructions": "email",
    "roles": [
      "pointOfContact",
      "originator"
    ]
  }
],
"created": "2018-08-19T08:10:00Z",
"updated": "2022-03-15T09:26:00Z",
"wmo:dataPolicy": "core",
"keywords": [
  "CLCH",
  "CLCL",
  "CLCM",
  "CLCT",
  "cloud",
  "MSLP",
  "Probability of temperature anomalies",
  "cape",
  "dew point",
  "geopotential height",
  "maximum temperature 2m",
  "minimum temperature 2m",
  "precipitation",
  "snow",
  "snowfall",
  "soil temperature",
  "temperature",
  "wind gusts 10m",
  "wind speed",
  "wind speed 10m"
],
"externalIds": [
  {
    "scheme": "DWD",
    "value": "de.dwd.icon-eps.ALL"
  }
],
"language": "eng",
"formats": [

```

```

    "GRIB2"
  ],
  "links": [
    {
      "rel": "download",
      "type": "text/html",
      "title": "Open Data Server DWD",
      "href": "https://opendata.dwd.de/weather/wmc/icon-eps/data/grib"
    },
    {
      "rel": "subscribe",
      "type": "application/json",
      "title": "Global Broker (Toulouse)",
      "href": "mqtt://[yourAccount]:[yourPassword]@wis2.dwd.de:8883/",
      "channel": "cache/a/wis2/deu/dwd-offenbach/data/core/weather/analysis-
prediction/forecast/model/#"
    }
  ]
}

```

Example: WIS2 Global Broker (service)

```

{
  "id": "urn:x-wmo:md:fra:meteo-france:global-broker",
  "type": "Feature",
  "conformsTo": [
    "http://wis.wmo.int/spec/wcmp/2/conf/core"
  ],
  "time": null,
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          -180,
          -90
        ],
        [
          -180,
          90
        ],
        [
          180,
          90
        ],
        [
          180,
          -90
        ],
        [

```

```

        -180,
        -90
    ]
]
},
"properties": {
    "type": "service",
    "title": "WIS2 Broker - M\u00e9t\u00e9o-France",
    "description": "TODO",
    "themes": [
        {
            "concepts": [
                {
                    "id": "weather",
                    "title": "Weather"
                },
                {
                    "id": "climate",
                    "title": "Climate"
                },
                {
                    "id": "hydrology",
                    "title": "Hydrology"
                },
                {
                    "id": "atmospheric-composition",
                    "title": "Atmospheric Composition"
                },
                {
                    "id": "cryosphere",
                    "title": "Cryosphere"
                },
                {
                    "id": "ocean",
                    "title": "Ocean"
                },
                {
                    "id": "space-weather",
                    "title": "Space Weather"
                }
            ],
            "scheme": "https://github.com/wmo-im/wcmp2-
codelists/blob/main/codelists/earth-system-discipline.csv"
        },
        {
            "concepts": [
                {
                    "id": "global-broker",
                    "title": "Global Broker"
                }
            ]
        }
    ]
}

```

```

        ],
        "scheme": "https://github.com/wmo-im/wcmp2-
codelists/blob/main/codelists/service-types.csv"
    },
    ],
    "contacts": [
        {
            "identifier": "meteo-france",
            "organization": "M\u00e9t\u00e9o-France",
            "positionName": "David Podeur",
            "emails": [
                {
                    "value": "gisc_support@meteo.fr"
                }
            ],
            "addresses": [
                {
                    "deliveryPoint": [
                        "42 Avenue Coriolis"
                    ],
                    "city": "Toulouse",
                    "postalCode": "31057",
                    "country": "France"
                }
            ],
            "links": [
                {
                    "type": "text/html",
                    "href": "https://donneespubliques.meteofrance.fr"
                }
            ],
            "contactInstructions": "email",
            "roles": [
                "host"
            ]
        }
    ],
    "created": "2022-11-11T11:00:00Z",
    "keywords": [
        "wis2",
        "global discovery catalogue",
        "metadata"
    ]
},
"links": [
    {
        "rel": "download",
        "type": "text/html",
        "title": "WMO WIS Global Cache - Deutscher Wetterdienst (Cached files <
24h during pilot)",
        "href": "https://opendata.dwd.de/test/wis2/cache"
    }
]

```

```

    },
    {
      "rel": "download",
      "type": "text/html",
      "title": "Open Data Server DWD (Metadata Archive)",
      "href": "https://opendata.dwd.de/test/wis2/cache_metadata"
    },
    {
      "rel": "related",
      "type": "application/json",
      "title": "WMO WIS2 Global Broker - M\u00e9t\u00e9o-France",
      "href": "mqtt://everyone:everyone@globalbroker.meteo.fr:8883",
      "channel": "cache/a/wis2/#",
      "distribution": {
        "maxMsgSize": 4096,
        "unit": "bytes"
      }
    },
    {
      "rel": "related",
      "type": "application/json",
      "title": "WMO WIS Global Broker - China Meteorological Administration",
      "href": "mqtt://everyone:everyone@gb.wis.cma.cn:1883",
      "channel": "cache/a/wis2/#",
      "distribution": {
        "maxMsgSize": 4096,
        "unit": "bytes"
      }
    }
  ]
}

```


Annex D: Codelists (Normative)

Annex E: Bibliography

- W3C/OGC: Spatial Data on the Web Best Practices, W3C Working Group Note 28 September 2017, <https://www.w3.org/TR/sdw-bp>
- W3C: Data on the Web Best Practices, W3C Recommendation 31 January 2017, <https://www.w3.org/TR/dwbp>
- W3C: Data Catalog Vocabulary, W3C Recommendation 16 January 2014, <https://www.w3.org/TR/vocab-dcat>
- IANA: Link Relation Types, <https://www.iana.org/assignments/link-relations/link-relations.xml>
- Linux Foundation: SPDX License List, <https://spdx.org/licenses>

Annex F: Revision History

Date	Release	Editor	Primary clauses modified	Description
2021-11-06	Template	Tom Kralidis	all	initial template