

WMO Core Metadata Profile (WCMP) Version 2

Table of Contents

1. Scope	3
2. Conformance	4
3. References	5
4. Terms and definitions	6
4.1. Abbreviated terms	6
5. Conventions	8
5.1. Identifiers	8
5.2. Examples	8
5.3. Schemas	8
5.4. Schema representation	8
5.4.1. Properties	8
5.5. Use of HTTPS	8
6. Introduction	10
6.1. Overview	10
6.1.1. WIS 2.0	10
6.1.2. Discovery metadata design considerations	11
6.1.3. Granularity	11
6.2. User stories	12
6.2.1. Domain specialist	12
6.2.2. Mass market	12
6.2.3. Developers	13
6.3. OGC API - Records - Part 1: Core	13
6.4. The WIS 2.0 Global Discovery Catalogue	13
6.5. Mass market considerations	14
7. The WMO Core Metadata Profile Record	15
7.1. Conformance Class Core	15
7.1.1. Overview	15
7.1.2. Validation	16
7.1.3. Granularity	17
7.1.4. Identifier	18
7.1.5. Conformance	19
7.1.6. Properties / Type	19
7.1.7. Properties / Title	20
7.1.8. Properties / Description	20
7.1.9. Properties / Keywords	20
7.1.10. Properties / Themes and Topic Hierarchy	21
7.1.11. Geospatial and Temporal Extents	22
7.1.12. Properties / Providers	25

7.1.13. Properties / Version	28
7.1.14. Properties / Persistent Identifiers	28
7.1.15. Properties / Record Creation Date	29
7.1.16. Properties / Record Update Date	30
7.1.17. Properties / WMO Data Policy	30
7.1.18. Distribution Information	33
7.1.19. Additional Properties	39
Annex A: Conformance Class Abstract Test Suite (Normative)	40
A.1. Conformance Class: Core	40
A.1.1. Validation	40
A.1.2. Identifier	40
A.1.3. Conformance	41
A.1.4. Type	42
A.1.5. Geospatial Extent	42
A.1.6. Temporal Extent	42
A.1.7. Title	43
A.1.8. Description	43
A.1.9. Themes and Topic Hierarchy	44
A.1.10. Providers	44
A.1.11. Record Creation Date	45
A.1.12. WMO Data Policy	45
A.2. Links	46
Annex B: Codelists	48
Annex C: Bibliography	49
Annex D: Revision History	50

World Meteorological Organization

Date: 2022-11-17

Version: 2.0.0-DRAFT-2022-11-17

Document location: TBD

Document status: DRAFT

Task Team on WIS Metadata (TT-WISMD) [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-imt/expert-team-metadata-0>]

Expert Team on Metadata Standards (ET-Metadata) [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-imt/et-metadata>]

Standing Committee on Information Management and Technology (SC-IMT) [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-imt>]

Commission for Observation, Infrastructure and Information Systems (INFCOM) [4: <https://community.wmo.int/governance/commission-membership/infcom>]

Copyright © 2022 World Meteorological Organization (WMO)

i. Abstract

Discovery and search are key components of WIS 2.0. 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: <https://ogcapi.ogc.org/records>].

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: <https://datatracker.ietf.org/doc/html/rfc7946>]) 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: <https://www.w3.org/TR/dwbp>] and Spatial Data on the Web Best Practices [8: <https://www.w3.org/TR/sdw-bp>] 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: https://en.wikipedia.org/wiki/FAIR_data].

ii. Keywords

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

wmo, wis 2.0, 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: https://library.wmo.int/doc_num.php?explnum_id=11113#page=9], 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 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 a WIS 2.0 catalogue.

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

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 codelists.

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 WIS 2.0 discovery catalogue 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.2 in more detail.

Chapter 3. References

- OGC: OGC 20-004, OGC API - Records - Part 1: Core 1.0 (2021) [12: <https://docs.ogc.org/DRAFTS/20-004.html>]
- OGC: OGC 17-069r, OGC API - Features - Part 1: Core 1.0 (2022) [13: <https://docs.opengeospatial.org/is/17-069r4/17-069r4.html>]
- IETF: RFC-7946 The GeoJSON Format (2016) [14: <https://datatracker.ietf.org/doc/html/rfc7946>]
- IETF: RFC-8259 The JavaScript Object Notation (JSON) Data Interchange Format the GeoJSON Format (2016) [15: <https://datatracker.ietf.org/doc/html/rfc8259>]
- W3C/OGC: Spatial Data on the Web Best Practices, W3C Working Group Note (2017) [16: <https://www.w3.org/TR/sdw-bp>]
- W3C: Data on the Web Best Practices, W3C Recommendation (2017) [17: <https://www.w3.org/TR/dwbp>]
- W3C: Data Catalog Vocabulary, W3C Recommendation (2014) [18: <https://www.w3.org/TR/vocab-dcat>]
- IANA: Link Relation Types (2020) [19: <https://www.iana.org/assignments/link-relations/link-relations.xml>]
- Linux Foundation: SPDX License List (2021) [20: <https://spdx.org/licenses>]
- IETF: JSON Schema (2022) [21: <https://json-schema.org>]
- WMO: WIS2 Topic Hierarchy (2022) [22: <https://github.com/wmo-im/wis2-topic-hierarchy>]
- OpenAPI Specification 3.1.0 (2022) [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

Abbreviation	Term
REST	Representational State Transfer
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/2.0>

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 2.0 schema can be found at <https://schemas.wmo.int/wcmp/2.0/wcmpRecordGeoJSON.yaml>

5.4. Schema representation

JSON Schema [24: <https://json-schema.org>] objects are used throughout this standard to define the structure of metadata records. These schema objects are also typically represented using YAML [25: <https://en.wikipedia.org/wiki/YAML>]. YAML is a superset of JSON, and in this standard are regarded as equivalent.

Metadata record instances are always defined 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.

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, authoritative groups 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 WIS 2.0 Principles, in order to support the WIS 2.0 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. WIS 2.0

WIS 2.0 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 WIS 2.0 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.

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

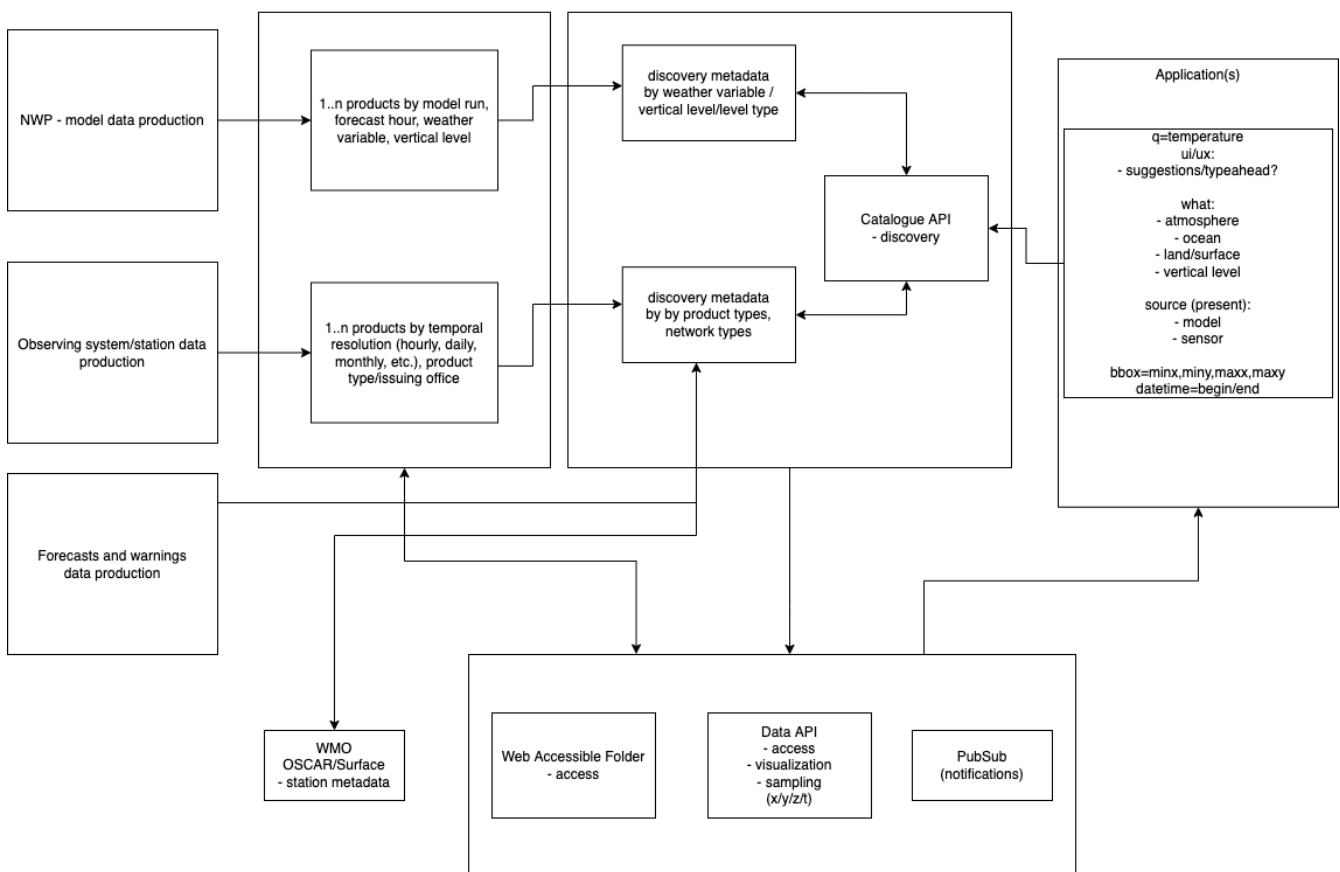


Figure 1. Discovery metadata workflow

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: <https://github.com/wmo-im/wcmp/issues/107>], the following user stories provide a description of features that are relevant to WIS 2.0 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 WIS 2.0 marketing video [27: https://gisc.dwd.de/wis2.0/WIS_2.0_final.mp4] 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 WIS 2.0, 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 or static records
- provides a core record model 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 WIS 2.0 Global Discovery Catalogue

The GDC will provide a central search endpoint, enabling users to traverse, browse and search data holdings in WIS 2.0. Key search predicate capabilities include:

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

Given the WIS 2.0 principles, use cases, OGC API - Records - Part 1: Core, and the WIS 2.0 Global Discovery Catalogue, WCMP provides a standards-based, clear and well-defined information model to facilitate the management and discovery of data within WIS 2.0.

6.5. Mass market considerations

Given WIS 2.0 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.

Chapter 7. The WMO Core Metadata Profile Record

A WCMP record provides descriptive information about a dataset made available through WIS 2.0.

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/2.0/req/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
<code>type</code>	required	A fixed value denoting the record as a GeoJSON Feature
<code>conformsTo</code>	required	The version of WCMP associated that the record conforms to
<code>properties.type</code>	required	The resource type described by the record
<code>properties.title</code>	required	A human-readable name for a given dataset

Property	Requirement	Description
<code>properties.description</code>	required	A free-text summary description of the dataset
<code>properties.keywords</code>	required	Keywords, tags, key phrases, or classification codes
<code>properties.themes</code>	required	Classifiers, categories and controlled vocabularies
<code>geometry</code>	required	Geospatial location associated with the dataset, in a geographic coordinate reference system
<code>properties.extent.spatial</code>	optional	Additional geospatial extents in other coordinate reference systems
<code>time</code>	required	Temporal extent associated with a dataset
<code>properties.extent.temporal</code>	optional	Additional time instants or periods
<code>properties.providers</code>	required	Contact information for the dataset
<code>properties.version</code>	optional	Version or edition of the dataset
<code>properties.externalIds</code>	optional	Persistent identifiers or handles for the dataset
<code>properties.recordCreated</code>	required	The date that the record was created
<code>properties.recordUpdated</code>	optional	The date that the record was updated
<code>properties.wmo:dataPolicy</code>	required	WMO data policy as per Resolution 1 (Cg-Ext(2021) [28: https://library.wmo.int/doc_num.php?explnum_id=11113#page=9]
<code>links</code>	required	Online linkages to data retrieval or additional resources associated with the dataset

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 2.0 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 2.0 schema.
B	Each WCMP record SHALL provide id , type , geometry and properties properties for GeoJSON compliance.
C	Each WCMP record type property SHALL be set to a fixed value of Feature for GeoJSON compliance.

7.1.3. Granularity

A WCMP record provides a description at the 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).

Recommendation 1	/rec/core/granularity
A	A WCMP record SHOULD provide a high level description of a dataset to facilitate simple information search queries.
B	A WCMP record SHOULD provide enough information to facilitate further interaction with a dataset (parameters, variables, spatiotemporal dimensions) as part of an associated service or capability.
C	A WCMP record SHOULD describe dataset parameters and variables as themes/concepts with an associated vocabulary Properties / Themes and Topic Hierarchy .
D	When dataset properties (parameters, variables, spatiotemporal extents) are made available via a data service or API, a WCMP record SHOULD provide templated links to facilitate easier data queries and filtering. See the Distribution Information section.

Examples

```
"themes": [  
  {  
    "concepts": [  
      "Temperature",  
      "Relative humidity",  
      "u-component of Wind",  
      "v-component of Wind"  
    ],  
    "scheme": "http://codes.wmo.int/grib2/codeflag/4.2"  
  }  
]
```

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

7.1.4. Identifier

A WCMP record utilizes the OARec **id** property to provide a unique identifier to a 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 id property.
B	A WCMP record identifier SHALL have the following notation: urn:x-wmo:md:{country}:{centre_id}:{local_identifier} .
C	A WCMP record identifier SHALL include a country 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"
```

7.1.5. Conformance

A WCMP record leverages the OGC API **conformsTo** property to identify the conformance classes that apply to the record. WCMP conformance identification is valuable for content/version detection and handling.

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/2.0"  
]
```

7.1.6. Properties / Type

WCMP records provide descriptive information about different resource types, such as datasets and their associated services.

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 codelist.

Example

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

7.1.7. Properties / Title

A WCMP record's `properties.title` property is a human-readable name for a given dataset.

Requirement 5	/req/core/title
A	A WCMP record SHALL provide a <code>properties.title</code> property.

Example

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

7.1.8. Properties / Description

A WCMP record has a `properties.description` property, which is a free-text summary description of the dataset.

Table 3. Example

Requirement 6	/req/core/description
A	A WCMP record SHALL provide a <code>properties.description</code> 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)."  
  ...  
}
```

7.1.9. Properties / Keywords

A WCMP record should have a `properties.keywords` property, typically represented using keywords, tags, key phrases, or classification codes.

Recommendation 2	/rec/core/keywords
------------------	--------------------

A	A WCMP record SHOULD provide a properties.keywords property, as a list of freeform text or tags.
---	---

Example

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

7.1.10. Properties / Themes and Topic Hierarchy

A WCMP record has a **properties.themes** property, which is a knowledge organization system used to classify the data. A WCMP record can have one or more themes. The WIS2 Topic Hierarchy is put forth as a specific theme/concept and is consistent with the topic as defined in distribution links advertising broker services.

Requirement 7	/req/core/themes_topic_hierarchy
A	A WCMP record SHALL provide a properties.themes property.
B	A WCMP record properties.theme SHALL provide ONE theme based on the WIS 2 Topic Hierarchy (additional themes are permitted based on other vocabularies).

Examples

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



```

"properties": {
  ...
  "themes": [
    {
      "concepts": [
        "wis2/CAN/eccc-msc/data/core/weather/surface-based-observations"
      ],
      "scheme": "https://github.com/wmo-im/wis2-topic-hierarchy"
    }
  ]
  ...
}

```

7.1.11. Geospatial and Temporal Extents

7.1.11.1. Geospatial Extent

WCMP records provide descriptive information regarding the geospatial properties of a dataset. Geospatial extents provide a useful indicator of the location of the dataset and facilitates searching the Global Discovery Catalogue with geospatial predicates.

Requirement 8	/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.

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
```

7.1.11.2. Properties / Additional Geospatial Extents

A WCMP record can communicate additional extents (for example, for non-geographic data or multiple temporal extents) via the `properties.extent` property. The `properties.extent.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 <code>properties.extent.spatial.bbox</code> item properties to express additional geospatial extents in other coordinate reference systems.

Example

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

7.1.11.3. Temporal Extent

WCMP records provide descriptive information regarding the temporal properties of a dataset. Temporal extents provide a useful indicator of the date/time period of the dataset and facilitates searching the Global Discovery Catalogue with temporal predicates.

Requirement 9	/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.

Recommendation 5	/rec/core/extent_temporal
A	For datasets with known or discrete intervals, a WCMP record SHOULD provide the temporal resolution (time.resolution) as a valid ISO 8601 duration (e.g., P1D).

Examples

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

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

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

```
"time": null
```

7.1.11.4. Properties / Additional Temporal Extents

A WCMP record can communicate additional time instants or periods via the `properties.extent.temporal` item property. Temporal extents can be fully bound or open in either direction (e.g., until or before). A `properties.extent.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>properties.extent.temporal</code> item properties to express additional temporal extents in other temporal reference systems.

Example

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

7.1.12. Properties / Providers

A WCMP record has one or more providers listed as part of the `properties.providers` property. These elements provide contact information based on the role of the provider.

Requirement 10	/req/core/providers
-----------------------	----------------------------

A	A WCMP record SHALL provide a <code>properties.providers</code> property.
B	A WCMP record <code>properties.providers</code> property SHALL provide at least TWO providers (as multiple provider objects or a single provider object with multiple roles) based on the metadata point of contact and the originator of the data. Providers are defined as either a URI or inline.

Examples

```
"properties": {
  ...
  "providers": [
    {
      "name": "Government of Canada; Environment and Climate Change Canada;
Meteorological Service of Canada",
      "positionName": "National Inquiry Response Team",
      "contactInfo": {
        "phone": {
          "office": "+01-819-997-2800"
        },
        "email": {
          "office": "enviroinfo@ec.gc.ca"
        },
        "address": {
          "office": {
            "deliveryPoint": "77 Westmorland Street, suite 260",
            "city": "Fredericton",
            "administrativeArea": "NB",
            "postalCode": "E3B 6Z4",
            "country": "Canada"
          },
          "onlineResource": {
            "href": "https://www.canada.ca/en/environment-climate-change.html"
          }
        },
        "hoursOfService": "0700h - 1500h EST",
        "contactInstructions": "email",
        "url": {
          "rel": "canonical",
          "type": "text/html",
          "href": "https://www.canada.ca/en/environment-climate-change.html"
        }
      },
      "roles": [
        {
          "name": "pointOfContact"
        },
        {
          "name": "originator"
        }
      ]
    }
  ]
  ...
}
```

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

```

"properties": {
  ...
  "providers": [
    "https://example.org/contact"
  ]
  ...
}

```

7.1.13. Properties / Version

Datasets may be versioned by an organization (version of an NWP model, processing chain/workflow, etc.), and data providers can make this information available when there are multiple versions of a dataset over time.

Permission 3	/per/core/version
A	A WCMP record MAY provide a properties.version property to describe the version of a given dataset.

Example

```

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

```

7.1.14. Properties / Persistent Identifiers

A persistent (or handle) identifier is used to provide a long lasting reference to a digital resource. Persistent identifiers are commonly used in scientific publications.

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

- Digital Object Identifiers (DOI) [29: <https://doi.org>]
- Archival Resource Key (ARK) [30: <https://arks.org>]
- Handle [31: <https://handle.net>]

Recommendation 6	/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 4	/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"
  } ]
  ...
}
```

7.1.15. Properties / Record Creation Date

A WCMP record has a `properties.recordCreated` property, which describes the date that the record was created.

Requirement 11	/req/core/record_creation_date
A	A WCMP record SHALL provide a <code>properties.recordCreated</code> property.

Example

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

7.1.16. Properties / Record Update Date

A WCMP record has a `properties.recordUpdated` property, which describes the date that the record was last updated.

Recommendation 7	/rec/core/record_update_date
A	A WCMP record SHOULD provide a <code>properties.recordUpdated</code> property when a record has been updated since its initial creation.

Example

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

7.1.17. Properties / WMO Data Policy

Based on the WMO Unified Data Policy for the International Exchange of Earth System Data (Resolution 1 (Cg-Ext(2021) [32: https://library.wmo.int/doc_num.php?explnum_id=11113#page=9], exchanged data are classified as `core` or `recommended`.

Requirement 12	/req/core/data_policy
A	A WCMP record SHALL provide exactly one <code>properties.wmo:dataPolicy</code> property.
B	A WCMP record's <code>properties.wmo:dataPolicy</code> property SHALL have exactly one <code>name</code> item which SHALL be either <code>core</code> or <code>recommended</code> .
C	If a WCMP record's <code>properties.wmo:dataPolicy.name</code> is equal to <code>recommended</code> , a <code>properties.wmo:dataPolicy.additionalConditions</code> item SHALL be provided.

Indicate additional conditions that apply when the resource is classified as **recommended**. Conditions on use are essential for the WMO Unified Data Policy, but for transparency and clarification any other conditions should also be indicated.

Requirement 13	/req/core/data_policy_conditions
A	A WCMP record's <code>properties.wmo:dataPolicy.additionalConditions</code> SHALL enumerate one to many objects to describe the conditions for the resource.
B	The <code>properties.wmo:dataPolicy.additionalConditions.name</code> property SHALL identify the condition applied to the dataset.
C	The <code>properties.wmo:dataPolicy.additionalConditions.scheme</code> property SHALL be used to identify a codelist or controlled vocabulary to which the condition name applies.
D	If there are no additional conditions, the fixed value of <code>none</code> SHALL be provided <code>properties.wmo:dataPolicy.additionalConditions.name</code> without any scheme identified.

Example: Core Data

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

Example: Recommended Data with no additional conditions

```
"properties": {
  ...
  "wmo:dataPolicy": {
    "name": "recommended",
    "additionalConditions": [{
      "name": "none"
    }]
  }
  ...
}
```

Example: Recommended Data with additional conditions

```
"properties": {  
  ...  
  "wmo:dataPolicy": {  
    "name": "recommended",  
    "additionalConditions": [{  
      "name": "copyright",  
      "scheme":  
        "https://standards.iso.org/iso/19139/resources/gmxCodelists.xml#MD_RestrictionCode"  
    }]  
  }  
  ...  
}
```

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

Recommendation 8	/rec/core/data_policy_conditions
A	A WCMP record's <code>properties.wmo:dataPolicy.additionalConditions</code> SHOULD provide additional provider-specific details to the conditions by adding an external URL in a <code>properties.wmo:dataPolicy.additionalConditions.link</code> object.
B	The <code>properties.wmo:dataPolicy.additionalConditions.scheme</code> property SHOULD be used for conditions which are based on a codelist of controlled vocabulary.
C	A WCMP record's <code>properties.wmo:dataPolicy.additionalConditions</code> SHOULD contain a <code>properties.wmo:dataPolicy.additionalConditions.link.title</code> property to provide additional human-readable information about the external URL.

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

```
"properties": {
  ...
  "wmo:dataPolicy": {
    "name": "recommended",
    "additionalConditions": [{
      "name": "license",
      "scheme":
"https://standards.iso.org/iso/19139/resources/gmxCodelists.xml#MD_RestrictionCode",
      "link": {
        "href": "https://www.eumetsat.int/eumetsat-data-licensing",
        "title": "EUMETSAT DATA LICENSING"
      }
    }]
  }
  ...
}
```

7.1.18. Distribution Information

7.1.18.1. Overview

A WCMP record provides information regarding how to access and retrieve data and products. This information is specified by defining a **links** array property, with one or more objects to corresponding data access services.

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 14	/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 WIS 2 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.
---	--

Recommendation 9	/rec/core/links
A	When a WCMP record provides a link object to a broker, the wmo:topic property SHOULD be consistent with the WIS2 topic as specified in a properties.themes object.

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" : "data",
    "type" : "application/json",
    "title": "WIS 2 notification service",
    "href" : "mqtt://broker.example.org",
    "wmo:topic": "cache/a/wis2/can/eccc-msc/data/core/weather/surface-based-observations"
  }
]
```

Permission 5	/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"
  }
]
```

Recommendation 10	/rec/core/distribution
A	A WCMP record SHOULD describe additional distribution information with the <code>properties.distribution</code> 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 <code>properties.distribution.availableFormat</code> property SHOULD describe the different formats that can be retrieved using the service link.
C	The <code>properties.distribution.availableFormat.description</code> SHOULD describe the associated format.
D	The <code>properties.distribution.availableFormat.typicalFilename</code> SHOULD describe the filenames that are available from the service link.
E	The <code>properties.distribution.availableFormat.typicalFileSize</code> SHOULD describe the typical filesize available from the service link.
F	The <code>properties.distribution.availableFormat.numberOfFiles</code> 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."
  }
}]
```

7.1.19. 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 6	/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"  
    ]  
  }  
  ...  
}
```

Annex A: Conformance Class Abstract Test Suite (Normative)

A.1. Conformance Class: Core

label

<http://www.wmo.int/spec/wcmp/2.0/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 **six** 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/2.0/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 valid **properties.type** property in the WCMP record.

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** value data types are integers or floats.

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

Check that **geometry** 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 RFC3339 representations, 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 and Topic Hierarchy

label

/conf/core/themes_topic_hierarchy

subject

/req/core/themes_topic_hierarchy

test-purpose

Validate that a WCMP record provides a themes property based on the WIS 2 Topic Hierarchy.

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

Within `properties.themes`, check for one theme with a scheme value of `http://www.wmo.int/spec/wis-topics/1.0`.

Within the theme based on the WIS2 Topic Hierarchy scheme, check that all values are valid topic hierarchies.

A.1.10. Providers

label

/conf/core/providers

subject

/req/core/providers

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.providers` property in the WCMP record.

Within `properties.providers`, check for the existence of `originator` and `pointOfContact` within the `roles` array property across all provider objects.

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 a `properties.recordCreated` 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 rights.

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

Check that `properties.wmo:dataPolicy` has an element `name` equal to `core` or `recommended`.

If `properties.wmo:dataPolicy.name` is equal to `recommended`, check for the existence of a `properties.wmo:dataPolicy.additionalConditions` element containing at least one object with a `name` property.

Check that `properties.wmo:dataPolicy.additionalConditions.conditions` is `none`. If the value is not `none`, then check for the existence of a `properties.wmo:dataPolicy.additionalConditions.scheme` property.

A.2. 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 **mqtt**s protocol, AND that **wmo:topic** 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: Codelists

Annex C: 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 D: Revision History

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