WMO WIS2 Notification Message Encoding

World Meteorological Organization

Date: 2024-03-12

Version: 1.0.0

Document location: TBD

Document status: DRAFT

Standing Committee on Information Management and Technology (SC-IMT)[1]

Commission for Observation, Infrastructure and Information Systems (INFCOM)[2]

Table of Contents

i. Abstract

WIS2 real-time data sharing is based on a message queuing protocol (MQP) supporting a publication/subscription (PubSub) mechanism. A user can subscribe to an MQP broker to receive real-time notifications of the existence of new data. The notification message received from the MQP broker contains a URL to download the data. In addition, the MQP broker offers a range of topics organised in a hierarchy. The users can select their topics of interest and subscribe to them to receive notifications and download data relevant to their work.

The standard notification message format ensures that the WIS2 ecosystem (data publisher, data user, and global services) is a robust, effective, and unified exchange platform for weather, climate, and water data.

This document defines the content, structure, and encoding for the WIS2 Notification Message Encoding. This standard is an extension of the OGC API - Features standard [3]. WIS2 Notification Message documents are provided as MQP payloads by WIS2 nodes, Global Broker services, as well as Replay API services (optional OGC API - Features services for data notifications).

WIS2 Notification Message documents shall be encoded in GeoJSON (RFC7946 [4]) as defined in this specification and shall be made available as MQP payloads. Additionally, they can be provisioned as defined by OGC API - Features.

Weather/climate/water data is by nature geospatial and temporal. The W3C Data on the Web Best Practices [5] and Spatial Data on the Web Best Practices [6] 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) [7].

ii. Keywords

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

wmo, wis 2.0, weather, climate, water, metadata, pubsub, mqp, message queuing protocol, GeoJSON

iii. Security Considerations

Based on the WMO Unified Data Policy for the International Exchange of Earth System Data (Resolution 1 (Cg-Ext(2021) [8], 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 a notification message published as part of a WIS2 Global Broker or a Replay API service.

The WIS2 Notification Message Encoding standard defined herein is an extension of the International Standard *GeoJSON*.

WIS2 Notification Message documents shall be encoded as GeoJSON as defined in *OGC API - Features - Part 1: Core*.

The primary purpose of WIS2 Notification Messages are to notify subscribers or clients of new data being published.

This specification defines the conformance requirements for the WIS2 Notification Message Encoding. Annex A defines the abstract test suite.

[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

[2] https://community.wmo.int/governance/commission-membership/infcom

[3] https://ogcapi.ogc.org/features

[4] https://datatracker.ietf.org/doc/html/rfc7946

[5] https://www.w3.org/TR/dwbp

[6] https://www.w3.org/TR/sdw-bp

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

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

Chapter 2. Conformance

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

GeoJSON provides an encoding for encoding geographic features (geometry, attributes). This standard is an extension of *GeoJSON*.

Data providers are required to comply with all conformance classes of this specification in support of providing MQP services.

WMO shall publish guidance material to assist data providers in constructing WIS2 Notification Messages.

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

The mandatory Requirements Class for this specification is:

- "WIS2 Notification Message Encoding Core"

Chapter 3. References

- OGC: OGC 17-069r, OGC API - Features - Part 1: Core 1.0.1 (2022) [1]
- IETF: RFC-7946 The GeoJSON Format (2016) [2]
- IETF: RFC-8259 The JavaScript Object Notation (JSON) Data Interchange Format (2017) [3]
- IETF: RFC 3339: Date and Time on the Internet: Timestamps (2002) [4]
- IETF: RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace (2005) [5]
- W3C/OGC: Spatial Data on the Web Best Practices, W3C Working Group Note (2017) [6]
- W3C: Data on the Web Best Practices, W3C Recommendation (2017) [7]
- W3C: Data Catalog Vocabulary, W3C Recommendation (2014) [8]
- IANA: Link Relation Types (2020) [9]
- IETF: JSON Schema (2022) [10]
- WMO: WIS2 Topic Hierarchy (2022) [11]

[1] https://docs.opengeospatial.org/is/17-069r4/17-069r4.html
[2] https://datatracker.ietf.org/doc/html/rfc7946
[3] https://datatracker.ietf.org/doc/html/rfc8259
[4] https://datatracker.ietf.org/doc/html/rfc3339
[5] https://datatracker.ietf.org/doc/html/rfc4122
[6] https://www.w3.org/TR/sdw-bp
[7] https://www.w3.org/TR/dwbp
[8] https://www.w3.org/TR/vocab-dcat
[9] https://www.iana.org/assignments/link-relations/link-relations.xml
[10] https://json-schema.org
[11] https://github.com/wmo-im/wis2-topic-hierarchy

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
DCPC	Data Collection and Production Centres
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
MQP	Message Queuing Protocol
MQTT	Message Queuing Telemetry Transport
NC	National Centre
NWP	Numerical Weather Prediction
OGC	Open Geospatial Consortium
PubSub	Publish / Subscribe
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universally Unique Identifier

Abbreviation	Term
W3C	World Wide Web Consortium
WCMP	WMO Core Metadata Profile
WIS	WMO Information System
WMO	World Meteorological Organization
WNM	WIS2 notification message

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/wnm/1

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/wnm/1.0/examples

5.3. Schemas

The WIS2 Notification Message schema can be found at https://schemas.wmo.int/wnm/1.0/notificationMessageGeoJSON.yaml

5.4. Schema representation

JSON Schema [1] objects are used throughout this standard to define the structure of metadata records. These schema objects are also typically represented using YAML [2]. 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"
```

[1] https://json-schema.org

[2] https://en.wikipedia.org/wiki/YAML

Chapter 6. Introduction

6.1. Motivation

MQP brokers provide "push" based services, supporting event-driven workflows, maximizing efficient use of network, bandwidth, and rapid response to time-sensitive events (e.g. severe weather events). However, MQP brokers do not specify a payload encoding. Using GeoJSON as a baseline for this specification provides broad interoperability, lowering the barrier and extending the reach of data in the WIS2 ecosystem and beyond.

6.2. Scenarios

The following scenarios are useful in understanding the drivers and principles that were used in the development of this specification:

6.2.1. Publish, Subscribe, Download

Event driven (PubSub) workflow involves a client connecting to a MQP broker, subscribing to one or more topics and receiving relevant notifications. Notifications provide information to ensure data integrity and spatiotemporal extents of a data granule.

6.2.2. Replay API workflow

API workflow involves a client connecting to an OGC API - Features services in order to query for messages in the past using spatial, temporal, or attribute predicates. A Replay API workflow is valuable in situations where connections to MQP brokers may drop, thereby allowing a given subscriber to gather past messages.

Chapter 7. The WIS2 Notification Message Encoding

A WIS2 Notification Message (WNM) provides descriptive information about data or metadata made available through WIS 2.0. The standard notification message format ensures that the WIS2 ecosystem (data publisher, data user, and global services) is a robust, effective, and unified data exchange platform for weather, climate, and water.

WIS2 Notification Message documents are provided as MQP payloads by WIS2 nodes, Global Broker services, as well as Replay API services.

A WNM provides notification metadata about data or metadata published to WIS2, including when the data was published, its spatial and temporal characteristics, and where to access and download.

WIS2 Notification Message is an extension of the OGC API - Features standard [3] and shall be encoded in GeoJSON (RFC7946 [4]).

7.1. Requirements Class "Core"

7.1.1. Overview

This Core Requirements Class provides requirements for a WIS2 Notification Message.

Requirements Class	
http://wis.wmo.int/spec/wnm/1/req/core	
Target type	Notification Metadata
Dependency	IETF RFC8259: The JavaScript Object Notation (JSON) Data Interchange Format
Dependency	JSON Schema
Dependency	GeoJSON (RFC7946)
Dependency	OGC API - Features - Core: Part 1
Pre-conditions	The record conforms to GeoJSON

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

Table 2. WNM core properties

Property	**Requirement**	**Description**
`id`	**required**	A universally unique identifier of the message (see Identifier)
`type`	**required**	A fixed value denoting the record as a GeoJSON `Feature` (see GeoJSON compliance)

Property	Requirement	Description
`version`	**required**	Version of message specification (see Version)
`geometry`	**required**	Geospatial location associated with the data or metadata (see Geometry)
`properties.pubtime`	**required**	The date and time of when the notification was published, in RFC3339 format (see Properties / Publication Time), UTC
`properties.data_id`	**required**	Unique identifier of the data as defined by the data producer (see Properties / Data Identification)
`properties.metadata_id`	optional	Identifier for associated discovery metadata record to which the notification applies (see Properties / Metadata identification)
`properties.producer`	optional	Identifies the provider that initially captured and processed the source data, in support of data distribution on behalf of other Members (see Properties / Producer)
`properties.datetime`	optional	Identifies the reference date and time of the data instance to which the notification is relayed, in RFC3339 format (see Properties / Temporal description), UTC
`properties.start_datetime`	optional	Identifies the start date and time of the data being published, in RFC3339 format (see Properties / Temporal description)
`properties.end_datetime`	optional	Identifies the end date and time of the data being published, in RFC3339 format (see Properties / Temporal description)

Property	Requirement	Description
`properties.cache`	optional	Indicates whether the data in the notification should be cached (if not specified, the default value is `true`) (see Properties / Cache)
`properties.integrity`	optional	Specifies a checksum to be applied to the data to ensure that the download is accurate (see Properties / Integrity)
`properties.content`	optional	Used to embed small products inline within the message (see Properties / Content)
`links`	**required**	Online linkages for data retrieval or additional resources associated with the dataset (see Links)

7.1.2. Message size

The WIS2 Notification Message allows for the transmission of messages in a compact manner and includes the ability to embed content inline as required (see Properties / Content).

Requirement 1	/req/core/message_size
A	A WNM message SHALL NOT exceed 8192 bytes.

7.1.3. GeoJSON compliance

The WIS2 Notification Message schema is based on *GeoJSON* (RFC7946) and its associated information model. Compliant messages are therefore compliant with *GeoJSON*.

Requirement 2	/req/core/validation
A	Each WNM SHALL validate without error against the WNM schema.
B	Each WNM SHALL provide `id`, `type`, `geometry` and `properties` properties for GeoJSON compliance.
C	Each WNM record `type` property SHALL be set to a fixed value of `Feature` for GeoJSON compliance.

7.1.4. Identifier

A universally unique identifier of the message using the UUID standard (RFC4122). The identifier is generated by the originator of the message. It provides the anti-loop feature that is needed to ensure that the message will be seen once by all Global Brokers. It remains the same throughout the

lifetime of the message in the WIS2 ecosystem.

The Properties / Data Identification is retained to ensure traceability and consistency of the same resource.

```
"id": "31e9d66a-cd83-4174-9429-b932f1abe1be"
```

Requirement 3	/req/core/identifier
A	The `id` property SHALL be a Universally Unique Identifier (UUID).

7.1.5. Version

The `version` property provides the version of WNM that the message conforms to.

Requirement 4	/req/core/version
A	A WNM SHALL provide information on version conformance via the `version` property.
B	The `version` property SHALL be fixed to `v04` for this version of the specification.

7.1.6. Geometry

The type of geometry in a notification message may be `Point` or `Polygon`. It can also be type `null` if the geometry cannot be derived.

Example: Point

```
{
  ...
  "geometry": {
    "type": "Point",
    "coordinates": [
      6.146255135536194,
      46.223296618227444
    ]
  }
  ...
}
```

Example: Point with elevation

```
{
  ...
  "geometry": {
    "type": "Point",
    "coordinates": [
      6.146255135536194,
```

```
      46.223296618227444,
      392
    ]
  }
  ...
}
```

Example: Polygon

```
{
  ...
  "geometry": {
    "type": "Polygon",
    "coordinates": [[
      [-7.75,40.43],
      [-7.75,78.46],
      [71.91,78.46],
      [71.91,40.43],
      [-7.75,40.43]
    ]]
  }
  ...
}
```

Example: null

```
{
  ...
  "geometry": null
  ...
}
```

Requirement 5	**/req/core/geometry**
A	A WNM record SHALL provide ONE geometry property to convey the geospatial properties of a notification using a geographic coordinate reference system (World Geodetic System 1984 [WGS 84]) and longitude and latitude decimal degree units.
B	The `geometry` property SHALL only provide one of a `Point` or `Polygon` geometry, or a `null` value when a geometry value is unknown or cannot be determined.

Permission 1	**/per/core/geometry**
A	The `geometry` property MAY provide a third element (height) as per clause 4 of the GeoJSON specification.

7.1.7. Properties / Publication Time

The `pubtime` property identifies the date/time when the notification was first posted or published by the originator. The date/time is encoded in RFC3339 format with the UTC timezone (`Z`).

The publication date/time is critical for subscribers to prevent message loss in providing awareness of how far behind the publisher they may be.

The `pubtime` property is also valuable for change detection as part of updates and deletion notifications.

Ensuring `pubtime` is properly managed for updates and deletions is important for data and metadata download workflows. For example, an out-of-date `pubtime` can lead to errors for clients when managing updates or deletions in their local copies of data. An update with newer `pubtime` and identical `datetime` indicates a newer version of the data or metadata.

```
"properties": {
  ...
  "pubtime": "2022-03-20T04:50:18.314854383Z"
  ...
}
```

Requirement 6	/req/core/pubtime
A	A WNM SHALL provide a `properties.pubtime` property.
B	The `properties.pubtime` property SHALL be in RFC3339 format.
C	The `properties.pubtime` property SHALL be in UTC timezone.
D	The `properties.pubtime` property SHALL be set to the current time by the original publisher of the notification.
E	The `properties.pubtime` property SHALL be set to the current time also for notifications about updates or deletions.
F	The `properties.pubtime` property SHALL NOT be modified by any intermediaries.

7.1.8. Properties / Data Identification

The `data_id` property uniquely identifies the data described by the notification and is defined by the data producer. A data producer may use an identification scheme of their choice.

```
"properties": {
  ...
  "data_id": "wis2/ma-marocmeteo/data/core/weather/surface-based-
observations/synop/WIGOS_0-504-1-60288_20240210T130000"
  ...
}
```

Requirement 7	/req/core/data_id
A	A WNM SHALL provide a `properties.data_id` property.
B	The `properties.data_id` property SHALL be unique within the scope of the relevant dataset.

Recommendation 2	/rec/core/data_id
A	The `properties.data_id` property SHOULD NOT use an opaque id. It should be encoded with meaningful values to support client-side filtering.

Permission 1	/per/core/data_id
A	The `properties.data_id` property MAY contain a valid WIS2 topic, without the channel and version.

7.1.9. Properties / Metadata identification

The `metadata_id` property uniquely identifies the associated discovery metadata record. This property is an important linkage between a WCMP2 dataset discovery metadata record and the related data notifications. The inclusion of this property allows a subscriber to consult additional documentation of the dataset and understand the access control applied to the data.

```
"properties": {
  ...
  "metadata_id": "urn:wmo:md:ca-eccc-msc:observations.swob"
  ...
}
```

Recommendation 3	/rec/core/metadata_id
A	A WNM SHOULD provide a `properties.metadata_id` property that identifies the associated WCMP2 dataset discovery metadata record. See requirement for metadata identification in WCMP2.

7.1.10. Properties / Producer

The `producer` property identifies the provider that initially captured and processed the source data, in support of data distribution on behalf of other Members.

```
"properties": {
  ...
  "producer": "fra"
  ...
}
```

Recommendation 4	/rec/core/producer
A	A WNM SHOULD provide a `properties.producer` property when publishing data on behalf of other Members.

7.1.11. Properties / Temporal description

The `datetime` property identifies the date and time of the data (for example, when a measurement was observed). When a data or metadata is updated or deleted, this value should identify the original data or metadata, which can be significantly different from the current time.

The `start_datetime` and `end_datetime` properties identify a temporal extent (for example, the start and end times of an NWP forecasting period).

All dates and times are encoded in RFC3339 format with the UTC timezone (`Z`).

A `null` value can also be used if a temporal description of the data cannot be derived.

Example: Temporal instant

```
"properties": {
  ...
  "datetime": "2022-03-20T04:45:00Z"
  ...
}
```

Example: Temporal extent

```
"properties": {
  ...
  "start_datetime": "2022-03-20T04:45:00Z",
  "end_datetime": "2022-03-22T04:45:00Z"
  ...
}
```

Example: No temporal description

```
"properties": {
  "datetime": null,
  ...
}
```

Requirement 8	/req/core/temporal
A	A WNM SHALL provide a temporal description by either a `properties.datetime` property or both the `properties.start_datetime` and `properties.end_datetime` properties.
B	The temporal description SHALL be in RFC3339 format.

C	The temporal description SHALL be in the UTC timezone.
D	The temporal description SHALL be set to `null` (using only `properties.datetime`) when a temporal description cannot be derived.

7.1.12. Properties / Cache

All core data, by default, is cached by Global Cache services.

However, a data producer can use the `properties.cache` value to request Global Cache services to not cache their `core` data granule.

Example: Specifying data not to be cached

```
"properties": {
  "cache": false,
  ...
}
```

Permission 2	**/per/core/cache**
A	A WNM MAY specify whether the data should be cached via the `properties.cache` property.

7.1.13. Properties / Integrity

For data verification, it is recommended to include data integrity information via the `integrity` property. Providing this information will allow data consumers to ensure that a given data granule has not been corrupted during download.

The `method` property provides a format of the hashing method used to enable an integrity check of the data. The preferred values are `sha256`, `sha384`, `sha512`, `sha3-256`, `sha3-384`, and `sha3-512`.

The `value` property provides the result of the hashing method in base64 encoding.

```
"properties": {
  ...
  "integrity": {
    "method": "sha512",
    "value":
"CPvTLiOfYRgfL3YNF/KKElwamwvLQwnzd96VnF2WoYuuH+hVIbwFSPQHHd/qa/fNVUBckviC5/HZs3Nx2jXEs
A=="
  }
  ...
}
```

Recommendation 5	**/rec/core/integrity**

A	A WNM SHOULD provide a `properties.integrity` property, consisting of a `method` property identifying the hashing method (`sha256`, `sha384`, `sha512`, `sha3-256`, `sha3-384`, `sha3-512`) and a `value` property of the hashing result, when it can be easily derived.

7.1.14. Properties / Content

The `content` property allows for the inclusion of data in the notification message when the encoded data is smaller than 4096 bytes. The limit takes into account the data encoding. That is, if the data are encoded in a form that changes the size, the resulting size must be less than 4096 bytes.

The `encoding` property provides the character encoding of the data (`UTF-8`, `Base64`, or `gzip`), the `gzip` encoding means that the data are compressed using algorithm defined in RFC1952 and consequently converted to text using Base64 encoding.

The `value` property provides the data in accordance with the `encoding` property.

The `size` property provides the size, in bytes, of the data in its original unencoded form, therefore this value shall not be directly compared with the aforementioned size limit.

```
"properties": {
  ...
  "content": {
    "encoding": "utf-8",
    "value": "encoded bytes from the file",
    "size": 457
  }
  ...
}
```

Requirement 9	**/req/core/content**
A	For data whose resulting size in the encoded form is greater than 4096 bytes, notifications SHALL NOT provide inline via `properties.content.value`. Note that the the encoding may both enlarge the data size, for example when binary data, e.g BUFR, is Base64 encoded, as well as reduce the size, for example when XML data are compressed with `gzip`.

Recommendation 6	**/rec/core/content**
A	A WNM SHOULD provide a `content` property, consisting of an `encoding` property (either `utf-8`, `base64`, or `gzip`), a `value` property of the data, as well as a `size` property with the length of the data.

Permission 3	**/per/core/content**

A	For data whose resulting size (after possible compression) is less than 4096 bytes, notifications MAY provide the data inline via `properties.content.value`.

7.1.15. Links

The `links` array property consists of one or more objects providing URLs to access data.

Each link object provides:

- an `href` property with a fully qualified link to access the data
- a `rel` property providing an IANA link relation [1] or WCMP2 defined extensions[2] describing the relationship between the link and the message
- a `type` property providing the media type of the data
- a `length` property providing the length (in bytes) indicating the size of the data
- a `security` property providing a description of the access control mechanism applied (i.e., for recommended data)

Links are used to communicate new data or metadata notifications. Links can also communicate when data or metadata has been deleted or invalidated.

Example: Canonical link

```
"links": [{
  "href": "https://example.org/data/4Pubsub/92c557ef-d28e-4713-91af-
2e2e7be6f8ab.bufr4",
  "rel": "canonical",
  "type": "application/x-bufr"
}]
```

Example: Multiple links

```
"links": [{
  "href": "https://example.org/data/4Pubsub/92c557ef-d28e-4713-91af-
2e2e7be6f8ab.bufr4",
  "rel": "canonical",
  "type": "application/x-bufr"
}, {
  "href": "https://example.org/oapi/collections/my-dataset/items/my-data-granule",
  "rel": "item",
  "type": "application/json"
}]
```

Requirement 10	**/req/core/links**
A	A WNM SHALL provide a `links` array property.

B	The `links` array property SHALL contain at least one link with, at a minimum, the `href` and `rel` properties.
C	The links for core data SHALL NOT require further action in order to download the resource.
D	The links SHALL be HTTP, HTTPS, FTP or SFTP.
E	For new data and metadata notifications, the `links` array property SHALL provide at least one link with an IANA link relation of `canonical` to clearly identify the preferred access link.
F	For data or metadata update notifications, the `links` array property SHALL provide at least one link with a link relation of `update` to clearly identify the preferred access link.
G	For data or metadata deletions, the `links` array property SHALL provide at least one link with a link relation of `deletion` to clearly identify data which has been deleted or removed.

Recommendation 7	**/rec/core/links**
A	A WNM SHOULD provide links using secure protocols such as HTTPS and SFTP, with HTTPS being the preferred option.
B	The `link` property SHOULD provide a `length` property to communicate the size of a given data download in advance of a data download workflow when the size of the data is known or can be easily derived.
C	The link relation of `deletion` SHOULD NOT be used for communicating a rolling data archive.

Permission 4	**/per/core/links**
A	A WNM `links` array property MAY provide link objects which reference APIs or Web Accessible Folders (WAF).

7.1.15.1. Access control

For recommended data, WNM links may also provide links to resources 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: Access controlled link

```
"links": [{
  "rel": "data",
  "type": "application/json",
  "title": "link to WAF endpoint",
  "href": "https://example.org/data/secure-data",
  "security": {
```

```
    "default": {
      "type": "http",
      "scheme": "basic",
      "description": "Please contact the data provider for accessing this secured
resource."
    }
  }
}]
```

7.1.16. Additional properties

A WIS2 Notification Message can be extended as required for organizational purposes by adding properties (of any type) in the message. Additional properties do not break compliance with this specification.

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

Permission 5	/per/core/additional_properties
A	A WNM MAY provide additional properties of any type in any part of the document as needed.

[1] https://www.iana.org/assignments/link-relations/link-relations.xhtml

[2] https://github.com/wmo-im/wcmp2-codelists/blob/main/codelists/link-type.csv

Annex A: Conformance Class Abstract Test Suite (Normative)

A.1. Conformance Class: Core

label

http://wis.wmo.int/spec/wnm/1/conf/core

subject

Requirements Class "core"

classification

Target Type:Notification Metadata

A.1.1. Message size

label

/conf/core/message_size

subject

/req/core/message_size

test-purpose

Validate that a WNM has a valid message size.

Check that the size of the complete WNM does not exceed 8192 bytes.

A.1.2. Validation

label

/conf/core/validation

subject

/req/core/validation

test-purpose

Validate that a WNM is valid to the authoritative WNM schema.

Run JSON Schema validation on the WNM against the WNM authoritative schema.

A.1.3. Identifier

label

/conf/core/identifier

subject

/req/core/identifier

test-purpose

Validate that a WNM has a valid identifier.

Check for the existence of an `id` property in the WNM.

Check that the `id` property is a valid UUID.

A.1.4. Version

label

/conf/core/identifier

subject

/req/core/identifier

test-purpose

Validate that a WNM has a valid version.

Check for the existence of an `version` property in the WNM.

Check that the `id` property is equal to `v04`.

A.1.5. Geometry

label

/conf/core/geometry

subject

/req/core/geometry

test-purpose

Validate that a WNM provides a valid geometry property.

Check for the existence of one `geometry` property in the WNM.

Check that all `geometry.coordinates` value data types are integers or floats.

Check that `geometry.coordinates` longitudinal values are between -180 and 180.

Check that `geometry.coordinates` latitudinal values are between -90 and 90.

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

A.1.6. Properties / Publication Time

label

/conf/core/pubtime

subject

/req/core/pubtime

test-purpose

Validate that a WNM has a valid pubtime.

Check for the existence of an `properties.pubtime` property.

Check that the `properties.pubtime` property is in RFC3339 format.

Check that the `properties.pubtime` property is in the UTC timezone.

A.1.7. Properties / Data Identification

label

/conf/core/data_id

subject

/req/core/data_id

test-purpose

Validate that a WNM has a valid data_id.

Check for the existence of an `data_id` property.

A.1.8. Properties / Temporal description

label

/conf/core/temporal

subject

/req/core/temporal

test-purpose

Validate that a WNM provides a valid temporal description.

Check for the existence of one `properties.datetime` or both `properties.start_datetime` and `properties.end_datetime`.

Check that the any of the `properties.datetime` or both `properties.start_datetime` and `properties.end_datetime` properties are in RFC3339 format.

Check that the any of the `properties.datetime` or both `properties.start_datetime` and `properties.end_datetime` properties are in the UTC timezone.

A.1.9. Links

label

/conf/core/links

subject

/req/core/links

test-purpose

Validate that a WNM provides a link array property.

Check for the existence of a single `links` array property in the WNM.

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

For each link object, check that the `href` property contains a valid protocol scheme of one of 'http', 'https', 'ftp', 'sftp'.

Check that the `links` array property contains one link object with a `rel` property with one of the values `canonical`, `update`, `deletion` as well as an `href` property.

Annex B: Schemas (Normative)

NOTE	The schema document will only be published on `schemas.wmo.int` once the standard has been approved.

B.1. WIS2 Notification Message Schema

```
$schema: https://json-schema.org/draft/2020-12/schema
$id: https://raw.githubusercontent.com/wmo-im/wis2-notification-
message/main/schemas/notificationMessageGeoJSON.yaml
title: WIS2 Notification Message Encoding
description: WIS2 Notification Message Encoding

type: object
properties:
  id:
    type: string
    format: uuid
    description: UUID (RFC4122) - Guarantee uniqueness of the message over (at least)
a 24h period.
  version:
    type: string
    description: Version of message specification.
    const: v04
  type:
    type: string
    enum:
      - Feature
  geometry:
    oneOf:
      - enum:
        - null
      - $ref: 'https://raw.githubusercontent.com/opengeospatial/ogcapi-
features/master/core/openapi/schemas/pointGeoJSON.yaml'
      - $ref: 'https://raw.githubusercontent.com/opengeospatial/ogcapi-
features/master/core/openapi/schemas/polygonGeoJSON.yaml'
  properties:
    type: object
    properties:
      pubtime:
        type: string
        format: date-time
        description: |
          Identifies the date/time of when the file was posted/published, in RFC3339
format.
          The publication date/time is critical for subscribers to prevent message
loss by knowing
          their lag (how far behind the publisher they are).
```

```
      data_id:
        type: string
        description: |
         Unique identifier of the data as defined by the data producer.
         Data producers SHOULD NOT use an opaque id, but something meaningful to
support client side filtering.
      metadata_id:
        type: string
        description: |
         Identifier for associated discovery metadata record to which the notification
applies to.
         The identification SHOULD be from a WCMP2 discovery metadata record
identifier from the WIS2
         Global Discovery Catalogue. Identifiers based on external catalogue systems
MAY be articulated as additional links.
      producer:
        type: string
        description: |
          Identifies the provider that initially captured and processed the source
data, in support of data distribution on behalf of other Members.
      datetime:
        type:
          - string
          - null
        format: date-time
        description: Identifies the date/time of the data being published, in RFC3339
format.
      start_datetime:
        type: string
        format: date-time
        description: Identifies the start date/time date of the data being published,
in RFC3339 format.
      end_datetime:
        type: string
        format: date-time
        description: Identifies the end date/time date of the data being published, in
RFC3339 format.
      cache:
        type: boolean
        description: |
          Whether the data in the notification should be cached (if not specified, the
default value is `true`).
          When set to `false`, WIS2 Global Cache services do not cache the canonical
link, and publish the
          notification with an unmodified canonical link (which points back to the
endpoint as specified by the data producer).
          The notification is always published by the Global Cache regardless to the
`cache` topic.
        default: true
      integrity:
        type: object
```

```
        description: Specifies a checksum to be applied to the data to ensure that the
download is accurate.
        properties:
          method:
            type: string
            description: |
                A specific set of methods for calculating the checksum algorithms:
                * ``sha256``: the Secure Hash Algorithm 2, 256 bits, value is base64
encoded.
                * ``sha384``: the Secure Hash Algorithm 2, 384 bits, value is base64
encoded.
                * ``sha512``: the Secure Hash Algorithm 2, 512 bits, value is base64
encoded.
                * ``sha3-256``: the Secure Hash Algorithm 3, 256 bits, value is base64
encoded.
                * ``sha3-384``: the Secure Hash Algorithm 3, 384 bits, value is base64
encoded.
                * ``sha3-512``: the Secure Hash Algorithm 3, 512 bits, value is base64
encoded.
            enum:
              - sha256
              - sha384
              - sha512
              - sha3-256
              - sha3-384
              - sha3-512
          value:
            type: string
            contentEncoding: base64
            description: Checksum value.
        required:
          - method
          - value
      content:
        type: object
        description: Used to embed small products inline within the message.
        properties:
          encoding:
            type: string
            description: Encoding of content
            enum:
              - utf-8
              - base64
              - gzip
          size:
            type: integer
            maximum: 4096
            description: |
              Number of bytes contained in the file. Together with the ``integrity``
property, it provides additional assurance that file content was accurately received.
              Note that the limit takes into account the data encoding used, including
```

```
data compression (for example `gzip`).
          value:
            type: string
            description: The inline content of the file.
            maxLength: 4096
        required:
          - encoding
          - size
          - value
    required:
      - pubtime
      - data_id
    oneOf:
      - allOf:
        - required:
          - start_datetime
          - end_datetime
      - allOf:
        - required:
          - datetime

  links:
    type: array
    minItems: 1
    items:
      allOf:
        - $ref: 'https://raw.githubusercontent.com/opengeospatial/ogcapi-
features/master/core/openapi/schemas/link.yaml'
        - properties:
            security:
              type: object
              $ref: https://raw.githubusercontent.com/OAI/OpenAPI-
Specification/3.1.0/schemas/v3.0/schema.yaml#/definitions/Components/properties/securi
tySchemes

required:
  - id
  - version
  - type
  - geometry
  - properties
  - links
example:
  $ref: https://raw.githubusercontent.com/wmo-im/wis2-notification-
message/main/examples/example1.json
```

Annex C: Examples (Informative)

C.1. WIS2 Notification Message Examples

Example: single observation including meaningful geometry, observation datetime and content


```
{
    "id": "31e9d66a-cd83-4174-9429-b932f1abe1be",
    "version": "v04",
    "type": "Feature",
    "geometry": {
        "type": "Point",
        "coordinates": [
            6.146255135536194,
            46.223296618227444
        ]
    },
    "properties": {
        "pubtime": "2022-03-20T04:50:18Z",
        "datetime": "2022-03-20T04:45:00Z",
        "integrity": {
            "method": "sha512",
            "value": "A2KNxvks...S8qfSCw=="
        },
        "data_id": "dataset/123/data-granule/UANT01_CWAO_200445___15103.bufr4",
        "metadata_id": "urn:wmo:md:ca-eccc-msc:observations.swob",
        "content": {
            "encoding": "utf-8",
            "value": "encoded bytes from the file",
            "size": 457
        }
    },
    "links": [
        {
            "href": "https://example.org/data/4Pubsub/92c557ef-d28e-4713-91af-
2e2e7be6f8ab.bufr4",
            "rel": "canonical",
            "type": "application/x-bufr"
        },
        {
            "href": "https://example.org/oapi/collections/my-dataset/items/my-data-
granule",
            "rel": "item",
            "type": "application/json"
        }
    ]
}
```

Example: NWP product including polygon `geometry`, `start_datetime` and `end_datetime`. The polygon is an array with 5 elements. The first (and last) elements are most southwestern point of the rectangle using longitude and latitude and followed by other coordinates in a clockwise direction.


```
{
    "id": "31e9d66a-cd83-4174-9429-b932f1abe1be",
    "version": "v04",
    "type": "Feature",
    "geometry": {
        "type": "Polygon",
        "coordinates": [
            [
                [
                    -7.75,
                    40.43
                ],
                [
                    -7.75,
                    78.46
                ],
                [
                    71.91,
                    78.46
                ],
                [
                    71.91,
                    40.43
                ],
                [
                    -7.75,
                    40.43
                ]
            ]
        ]
    },
    "properties": {
        "pubtime": "2022-03-20T04:50:18Z",
        "start_datetime": "2022-03-20T00:06:00Z",
        "end_datetime": "2022-03-20T00:12:00Z",
        "integrity": {
            "method": "sha512",
            "value": "A2KNxvks...S8qfSCw=="
        },
        "data_id": "dataset/4546/abcdef-abcd-abcd-abcd-abcdefabcd",
        "metadata_id": "urn:wmo:md:fr-meteo-france:gap123"
    },
    "links": [
        {
            "href": "https://example.org/data/nwp/92c557ef-d28e-4713-91af-
2e2e7be6f8ab.grib",
            "rel": "canonical",
```

```
            "type": "application/x-grib2"
        }
    ]
}
```

Example: minimal viable message compliant to the specification

```
{
    "id": "31e9d66a-cd83-4174-9429-b932f1abcdef",
    "version": "v04",
    "type": "Feature",
    "geometry": null,
    "properties": {
        "pubtime": "2022-11-20T16:40:37Z",
        "datetime": "2022-11-11T11:11:11Z",
        "data_id": "data/data-123/items/abcdefab-abcd-1234-5678-0123456789ab",
        "metadata_id": "urn:wmo:md:test-nmhs-xyz:some_dataset",
        "cache": false
    },
    "links": [
        {
            "href": "https://example.org/92c557ef-d28e-4713-91af-2e2e7be6f8ab.txt",
            "rel": "canonical"
        }
    ]
}
```

Example: notification of the deletion of a data granule

```
{
    "id": "31e9d66a-cd83-4174-9429-b932f1abcdef",
    "version": "v04",
    "type": "Feature",
    "geometry": null,
    "properties": {
        "pubtime": "2022-12-22T16:40:37Z",
        "datetime": "2022-11-11T11:11:11Z",
        "data_id": "data/data-123/items/abcdefab-abcd-1234-5678-0123456789ab",
        "metadata_id": "urn:wmo:md:test-nmhs-xyz:some_dataset",
        "cache": false
    },
    "links": [
        {
            "href": "https://example.org/92c557ef-d28e-4713-91af-2e2e7be6f8ab.txt",
            "rel": "deletion"
        }
    ]
}
```

Annex D: 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
- IANA: Link Relation Types, https://www.iana.org/assignments/link-relations/link-relations.xml

Annex E: Revision History

Date	Release	Editor	Primary clauses modified	Description
2022-12-05	Template	Tom Kralidis	all	update from TT-Protocols/ET-W2AT