

WIS2 Cookbook

World Meteorological Organization

Date: 2025-04-07

Version: 2024-04-16

Document status: DRAFT

Document location: <https://wmo-im.github.io/wis2-cookbook/cookbook/wis2-cookbook-DRAFT.html>

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

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

Copyright © 2024 World Meteorological Organization (WMO)

Table of Contents

1. Introduction	3
1.1. Purpose	3
1.2. Key documents	3
2. Recipes for data consumers	4
2.1. Searching the Global Discovery Catalogue	4
2.1.1. Spatial queries	4
2.1.2. Temporal queries	4
2.1.3. Equality queries	4
2.1.4. Freetext search	4
2.1.5. Sorting	5
2.1.6. Paging	5
2.2. Finding data subscription services from the Global Discovery Catalogue	5
3. Recipes for data publishers	7
3.1. Validating a WIS2 Notification Message	7
3.2. Publishing a WIS2 Notification Message with access control	7
3.3. Publishing a WIS2 Notification Message with embedded data	9
3.4. Publishing a WIS2 Notification Message for resource deletion	11
3.5. Validating a WMO Core Metadata Profile record	12
3.6. Advertising client side filters for data subscriptions in WCMP2 and WNM	14
3.6.1. Example: Surface weather observations	15
3.6.2. Example: Numerical weather prediction based forecast	16
4. Recipes for Earth system discipline domain experts	19
4.1. Defining and proposing topic for WMO Earth system disciplines	19
4.1.1. Core levels	19
4.1.2. Additional Earth domain specific topics levels	19
4.1.3. Dos and don'ts when defining Earth system discipline topics	20
5. Recipes for Global Service operators	23

Chapter 1. Introduction

1.1. Purpose

In conjunction with the Manual on the WMO Information System volume II (WMO-No. 1060) (Manual on WIS volume II: WIS 2.0), the present Guide to the WMO Information System volume II (Guide to WIS volume II: WIS 2.0) is designed to ensure adequate uniformity and standardization in the data, information and communication practices, procedures and specifications employed by Members of the World Meteorological Organization (WMO) in the operation of the WMO Information System WIS 2.0 as it supports the mission of the Organization. The Manual on WIS contains standard and recommended practices, procedures and specifications. The Guide to WIS contains additional information concerning practices, procedures, and specifications that Members are invited to follow or implement in establishing and conducting their arrangements in compliance with the WMO Technical Regulations and in developing meteorological and hydrological services.

This cookbook provides various code snippets, recipes and workflow examples in support of WIS2 requirements. This cookbook is a working document; contributions are encouraged and can be added via [GitHub](#).

1.2. Key documents

WIS2 Manuals and Guides can be found at the following locations:

- [Manual on the WMO Information System](#)
- [Guide to the WMO Information System Volume II - WMO Information System 2.0](#)
- [Provisions for the Transition from the WMO Information System \(WIS\) 1.0 and Global Telecommunication System to WIS 2.0](#)

[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>

Chapter 2. Recipes for data consumers

2.1. Searching the Global Discovery Catalogue

The [Global Discovery Catalogue \(GDC\)](#) allows for a wide range of query predicates to search for data in WIS2 as per the OGC API - Records - Part 1: Core specification.

The GDC can be searched via the `/collections/wis2-discovery-metadata/items` endpoint. This endpoint provides a number query parameters as described in the examples below.

NOTE: examples below are not URL encoded for clarity / readability, but should be when interacting with the GDC.

2.1.1. Spatial queries

- search for metadata records of data in Canada: `bbox=-142,42,-52,84`

Note that the format of `bbox` is comma-separated values in the following order:

- minimum longitude
- minimum latitude
- maximum longitude
- maximum latitude

2.1.2. Temporal queries

- search for metadata records updated since 29 July 2024: `datetime=2024-07-29/..`
- search for metadata records updated before 29 July 2024: `datetime=../2024-07-29`
- search for metadata records updated on 29 July 2024: `datetime=2024-07-29`

2.1.3. Equality queries

- search for metadata records whose title contains the terms hourly observations: `title=hourly observations`
- search for metadata records whose title contains the terms hourly or observations: `title=hourly | observations`
- search for metadata records for a specific contact organization `contacts.organization=Direction Generale de la Météorologie`

2.1.4. Freetext search

- search metadata records for temperature: `q=temperature`
- search metadata records for GRIB2 data: `q=GRIB2`
- search metadata records for any GRIB data: `q=*GRIB*`

- search metadata records for any GRIB data in Germany: `q=*GRIB* AND germany`
- search for either GRIB data or data in Europe with subscriptions to the Météo-France Global Broker: `q=(*GRIB* OR *Europe*) AND *globalbroker*`
- search for data from Belize with MQTT subscription capabilities: `q="cache/a/wis2/bz-nms"`

2.1.5. Sorting

- sort search results by title, ascending: `sortby=title`
- sort search results by title, descending: `sortby=-title`

2.1.6. Paging

- present search results 1-10: `limit=10`
- present search results 11-20: `limit=10&offset=10`
- limit to 3 search results: `limit=3`

2.2. Finding data subscription services from the Global Discovery Catalogue

The [Global Discovery Catalogue \(GDC\)](#) contains both real-time and non real-time data. The [WMO Core Metadata Profile \(WCMP2\)](#) allows for description of real-time data via its [distribution information](#), which data publishers use to describe and define connectivity and subscription information for a given dataset.

A typical WCMP2 distribution link for data subscriptions can be found below:

```
{
  "rel": "items",
  "href": "mqtt://everyone:everyone@globalbroker.meteo.fr:8883",
  "channel": "origin/a/wis2/ca-eccc-msc/data/core/hydrology",
  "type": "application/geo+json",
  "title": "Data notifications"
}
```

NOTE

The `channel` property represents WIS2 topic which can be used to subscribe to the `href` property (i.e. the MQTT address) of the Global Broker (GB).

Programmatically, a GDC client can query the catalogue and filter the results for real-time subscriptions in the following manner:

```
import requests

response = requests.get('https://wis2-gdc.weather.gc.ca/collections/wis2-discovery-metadata/items').json()
```

```

def is_wis2_subscription_link(link) -> bool:
    if (link['href'].startswith('mqtt') and
        link.get('channel', '').startswith(('origin/a/wis2', 'cache/a/wis2'))):
        return True

for feature in response['features']:
    for link in feature['links']:
        if is_wis2_subscription_link(link):
            print('WIS2 subscription link')

```

Using the **href** and **channel** properties of a matching link object, a client can connect and subscribe to data notifications for a given dataset.

Chapter 3. Recipes for data publishers

3.1. Validating a WIS2 Notification Message

A [WIS2 Notification Message](#) provides a [JSON Schema](#) which can be used by any programming language that supports JSON and JSON Schema validation.

Using Python and check-jsonschema

```
# install check-jsonschema Python Package from the Python Package Index (PyPI)
pip3 install check-jsonschema

# download WNM schema
curl -O http://schemas.wmo.int/wnm/1.0.0/schemas/wis2-notification-message-bundled.json

# run schema validation
check-jsonschema --schemafile wis2-notification-message-bundled.json
/path/to/my/wnm.json
```

The [pywis-pubsub](#) tool provides a test suite to validate a message against the WNM specification requirements, as well as a Python API for application integration. Consult the [pywis-pubsub README](#) on GitHub for more information/examples.

Using pywis-pubsub

```
# install pywis-pubsub
pip3 install pywis-pubsub

# sync WIS2 notification schema
pywis-pubsub schema sync

# validate WNM against abstract test suite (file on disk)
pywis-pubsub ets validate /path/to/file.json

# validate WNM against abstract test suite (URL)
pywis-pubsub ets validate https://example.org/path/to/file.json
```

3.2. Publishing a WIS2 Notification Message with access control

Recommended data in WIS2 may be open or access controlled. For data publication with access control implications, WNM provides a [security](#) object as part a link object. The [security](#) object is defined using [OpenAPI Security Scheme definitions](#).

Access control using HTTP Basic authentication

```
{
  "rel": "canonical",
  "type": "application/grib2",
  "href": "https://example.org/my/protected/data/nwp/12/003/20240805120000-air-temp-500.grib2",
  "security": {
    "default": {
      "type": "http",
      "scheme": "basic",
      "description": "Please contact us for access information"
    }
  }
}
```

Access control using an API key

```
{
  "rel": "canonical",
  "type": "application/geo+json",
  "href": "https://example.org/my/protected/data/nwp/12/003/20240805120000-air-temp-500.grib2",
  "security": {
    "default": {
      "type": "apiKey",
      "name": "api-key",
      "in": "query",
      "description": "Please see https://example.org/contact-us for more information"
    }
  }
}
```

Note:

- the child property under **security** (**default** in the examples above) can be any text or label. We use **default** here as a convention
- the **security.default.name** is the name of the API key parameter as defined by your API service
- only properties defined in the OpenAPI Security Scheme definition are allowed. Any additional properties will invalidate the WNM

Of course, always ensure your WNM is valid (see [\[Validate a WIS2 Notification Message\]](#) for more information).

3.3. Publishing a WIS2 Notification Message with embedded data

A [WIS2 Notification Message](#) provides the ability to embed data as part of a message (see the [Properties / Content](#) section of the specification).

Providing embedded data for direct access offers users access to the data without the need for downloading the file via the notification link (see the [Links](#) section of the specification).

In some cases, a data publisher also may not wish to manage or publish the data link to infrastructure to an HTTP or FTP server.

When the data publisher decides to use the embedded data feature without the need of implementing an actionable link providing access to the same data, the data publisher will:

- use `properties.content` object to provide the data
- use `properties.cache=false` to instruct WIS2 Global Cache Services to not cache the data
- provide a link object whose `href` results in HTTP 204 (No Content)

To provide an HTTP 204 link, there a couple of options:

Option 1: configure your web server to provide an endpoint that always returns 204:

Nginx example configuration

```
# always return HTTP 204 when accessing /no-content-here
location /no-content-here/ {
    return 204;
}
```

Apache example configuration

```
# always return HTTP 204 when accessing /no-content-here
Redirect 204 no-content-here
```

which then would result in the link object as follows:

```
{
  "href": "https://example.org/no-content-here",
  "rel": "canonical"
}
```

Option 2: use a service that provides HTTP responses as a service. One such service is the [HTTP Response API](#), which then would result in the link object as follows:

```
{
```

```
"href": "https://http.codes/204",  
"rel": "canonical"  
}
```

Embedded data example:

```
{  
  "id": "31e9d66a-cd83-4174-9429-b932f1abe1be",  
  "conformsTo": [  
    "http://wis.wmo.int/spec/wnm/1/conf/core"  
  ],  
  "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=="  
    },  
    "cache": false,  
    "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://http.codes/204",  
      "rel": "canonical"  
    }  
  ]  
}
```

For cases of publishing recommended data, the same approaches/example can be used, without the need to set **properties.cache**.

3.4. Publishing a WIS2 Notification Message for resource deletion

A [WIS2 Notification Message](#) provides the ability to publish notifications for new, updated or deleted data and metadata (see the [Links](#) section of the specification).

Similar to the [embedded data Recipe](#), for data or metadata deletions (core or recommended data), a data publisher may not wish to manage or publish the data or metadata link to infrastructure to an HTTP or FTP server.

In this case, a similar strategy can be used to provide an HTTP 204 No Content link, as per the [embedded data Recipe](#), along with setting `rel=deletion` in the link object.

Resource deletion example:

```
{
  "id": "31e9d66a-cd83-4174-9429-b932f1abe1be",
  "conformsTo": [
    "http://wis.wmo.int/spec/wnm/1/conf/core"
  ],
  "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=="
    },
    "cache": false,
    "data_id": "dataset/123/data-granule/UANT01_CWAO_200445___15103.bufr4",
    "metadata_id": "urn:wmo:md:ca-eccc-msc:observations.swob",
  },
  "links": [
    {
      "href": "https://http.codes/204",
      "rel": "deletion"
    }
  ]
}
```

3.5. Validating a WMO Core Metadata Profile record

The [pywcmp](#) tool provides a test suite to validate a message against the WCMP2 specification requirements, as well as a Python API for application integration. Consult the [pywcmp README](#) on GitHub for more information/examples.

Using pywcmp

```
# install pywcmp
pip3 install pywcmp

# sync WCMP2 schemas and codelists
pywcmp bundle sync

# validate WCMP2 against abstract test suite (file on disk)
pywcmp ets validate /path/to/file.json

# validate WCMP2 against abstract test suite (URL)
pywcmp ets validate https://example.org/path/to/file.json
```

A WCMP2 record can also be validated using pywcmp "as a service" using the Canadian WIS2 Global Discovery Catalogue, which provides an online validator:

- Navigate to <https://wis2-gdc.weather.gc.ca/openapi?f=html>
- Navigate to section **pywcmp-wis2-wcmp2-ets**, endpoint `/processes/pywcmp-wis2-wcmp2-ets/execution` (POST)
- Click "Try it out"
- In the section "Mandatory execute request JSON", paste the WCMP2 JSON inside the **record** object

pywcmp-wis2-wcmp2-ets

GET

/processes/pywcmp-wis2-wcmp2-ets

Get process metadata

POST

/processes/pywcmp-wis2-wcmp2-ets/execution

Process WCMP2 ETS validator execution

Validate a WCMP2 document against the ETS

Parameters

Cancel

Reset

No parameters

Request body

required

application/json

Mandatory execute request JSON

```
{
  "inputs": {
    "fail_on_schema_validation": true,
    "record": {
      "id": "urn:wmo:md:ca-eccc-msc:weather.observations.swob-realtime",
      "conformsTo": [
        "http://wis.wmo.int/spec/wcmp/2/conf/core"
      ],
      "time": {
        "interval": [
          "2010-11-11T11:11:11Z",
          "..."
        ]
      }
    },
    "type": "Feature",
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [
          r
        ]
      ]
    }
  }
}
```

Execute

Clear

Figure 1. WIS2 GDC online validator, request

- Click "Execute"

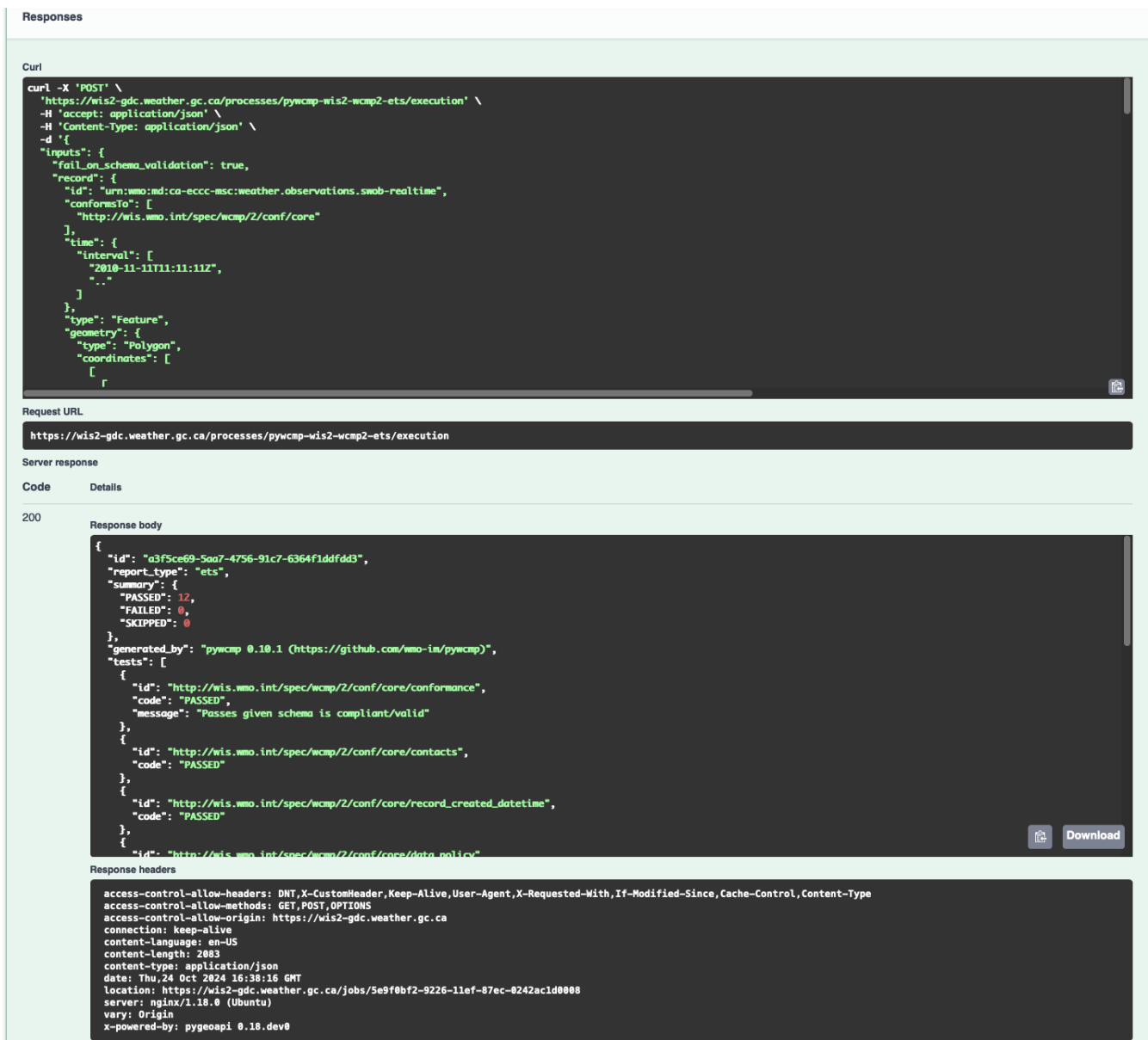


Figure 2. WIS2 GDC online validator, response

A response will be provided with validation results.

3.6. Advertising client side filters for data subscriptions in WCMP2 and WNM

A key concept of a WCMP2 record is "actionable links"; this means being able to access a dataset or data granule without any further interactions. For real-time data, a WCMP2 record provides linkages to the WIS2 Global Broker via the MQTT protocol. At its core, MQTT has two key components:

- topic: the topic to subscribe to
- message payload: the message provided as part of a notification to a given topic

WIS2 defines the WIS2 Topic Hierarchy (WTH) and WIS2 Notification Message (WNM) standards which provide a standards-based GeoJSON payload/message.

A typical MQTT link in a WCMP2 document is defined as follows:

Typical WCMP2 MQTT link

```
{
  "rel" : "items",
  "type" : "application/geo+json",
  "title": "WIS2 notification service",
  "href" : "mqtt://example.org",
  "channel": "cache/a/wis2/ca-eccc-msc/data/core/weather/surface-based-
observations/synop"
}
```

Given WCMP2, WTH and WNM, a user can subscribe to topics related to data of interest for download and access.

In some cases, a dataset may be organized in a manner which requires additional further "filtering" such that a data consumer is only interested in a certain subset of the data granules being advertised by a given WNM. Some examples include (but are not limited to), where a data consumer may be only be interested in:

- surface weather observations from a certain station, or
- numerical weather prediction forecast data for a certain timestep or weather parameter

To implement this behaviour, add additional properties to both WCMP2 and WNM as follows:

3.6.1. Example: Surface weather observations

Surface weather observations: WCMP2 MQTT link with additional properties

```
{
  "rel" : "items",
  "type" : "application/geo+json",
  "title": "Real-time notifications",
  "href" : "mqtt://globalbroker.meteo.fr:8883",
  "channel": "cache/a/wis2/ca-eccc-msc/data/core/weather/surface-based-
observations/synop",
  "properties": {
    "wigos_station_identififier": {
      "type": "string",
      "title": "WIGOS station identifier"
    }
  }
}
```

Surface weather observations: WNM additional properties

```
{
  "properties": {
```



```

    "wigos_station_identifier": "0-20000-0-71628"
  }
}

```

When implemented by a data producer, a data consumer can:

- subscribe to real-time notifications to the given topic
- perform client side filtering by against all incoming WNM with `properties.wigos_station_identifier = "0-20000-0-71628"`

3.6.2. Example: Numerical weather prediction based forecast

Numerical weather prediction: WCMP2 MQTT link with additional properties

```

{
  "rel" : "items",
  "type" : "application/geo+json",
  "title": "Real-time notifications",
  "href" : "mqtt://globalbroker.meteo.fr:8883",
  "channel": "origin/a/wis2/ca-ecce-msc/data/core/weather/prediction/forecast/medium-
range/deterministic/global",
  "properties": {
    "model_run": {
      "type": "string",
      "title": "Model run",
      "enum": [
        "00",
        "12"
      ],
      "example": "00"
    },
    "forecast_hour": {
      "type": "string",
      "title": "Forecast hour",
      "example": "004"
    }
  }
}

```

Numerical weather prediction: WNM additional properties

```

{
  "properties": {
    "model_run": "00",
    "forecast_hour": "004"
  }
}

```

A data producer would extend WCMP2 and WNM as follows:

- WCMP2: add a link **properties** object for MQTT links, where each key of the link **properties** object is a [JSON Schema property definition](#)
- WNM: add additional properties (key: value pairs) in the **properties** object as desired

When implemented by a data producer, a data consumer can:

- subscribe to real-time notifications to the given topic
- perform client side filtering against all incoming WNM with **properties.model_run** = "00" and **properties.forecast_hour** = "004"

A sample Python script can be found below. The script connects to the Météo-France Global Broker, subscribed to weather notifications from Environment and Climate Change Canada, Meteorological Service of Canada. The script then performs client side filtering by evaluating (for each WNM) the **properties.wigos_station_identifier** value to match a particular station (0-20000-0-71628).

Sample Python script to perform client side filtering

```
import json
from paho.mqtt import client as mqtt_client

broker = 'globalbroker.meteo.fr'
port = 8883
username = 'everyone'
password = 'everyone'
topic = 'cache/a/wis2/ca-eccc-msc/data/core/weather/surface-based-observations/synop'

wsi_to_filter = '0-20000-0-71628'

def connect_mqtt() -> mqtt_client:
    def on_connect(client, userdata, flags, reason_code, properties):
        if reason_code == 0:
            print(f'Connected to {broker}')
        else:
            print(f'Failed to connect: {reason_code}')

    def on_log(client, userdata, level, message):
        print("LOG:", message)

    client = mqtt_client.Client(mqtt_client.CallbackAPIVersion.VERSION2,
                                client_id='s123')
    client.username_pw_set(username, password)
    client.on_connect = on_connect
    client.on_log = on_log
    client.tls_set(tls_version=2)
    client.connect(broker, port)

    return client
```

```

def subscribe(client: mqtt_client):
    def on_message(client, userdata, message):
        message_dict = json.loads(message.payload.decode())

        print('Performing client side filtering')
        wsi = message_dict['properties'].get('wigos_station_identifier')

        if wsi != wsi_to_filter:
            print(f'Topic: {message.topic}')
            print(f'Payload: {message.payload.decode()}')

    client.subscribe(topic)
    client.on_message = on_message

def run():
    client = connect_mqtt()
    subscribe(client)
    client.loop_forever()

if __name__ == '__main__':
    run()

```

Chapter 4. Recipes for Earth system discipline domain experts

4.1. Defining and proposing topic for WMO Earth system disciplines

The [WIS2 Topic Hierarchy](#) describes the topic structure and levels to be used when publishing WIS2 notification messages.

The purpose of this document is to provide guidelines to domain experts so that the Topic Hierarchy definition is consistent and useful to address the needs of WIS2 users.

4.1.1. Core levels

Core topics are defined in the first 7 levels and address all Earth system disciplines in a consistent manner.

Core topic examples:

- `cache/a/wis2/ma-marocmeteo/core/data/weather`
- `origin/a/wis2/de-dwd/core/data/ocean`
- `cache/a/wis2/jp-jma/core/data/weather`

The Earth system disciplines [defined](#) by WTH are as follows:

- `atmospheric-composition`
- `climate`
- `cryosphere`
- `hydrology`
- `ocean`
- `space-weather`
- `weather`

Topics within each Earth system discipline are then defined by domain experts, reviewed, approved, and published by WMO.

4.1.2. Additional Earth domain specific topics levels

4.1.2.1. What is the use of the Topic Hierarchy ?

The definition of the Topic Hierarchy is heavily linked to the use of Publish-Subscribe (Pub/Sub) protocols, here MQTT[S], in WIS2. Users, when subscribing to the Global Broker, can decide which notification messages they wish to receive.

4.1.2.2. MQTT wildcards

For example, connecting to a Global Broker and subscribing to:

`cache/a/wis2/+/core/data/weather/surface-weather-observation/synop`

users will receive a notification when a new synop is made available from **any** centre-id publishing message on WIS2. The **+** character is a single level wildcard for MQTT subscription.

A user can also choose to subscribe to:

`cache/a/wis2/de-dwd/core/data/weather/#`

In this case, users will receive notification messages from the **de-dwd** centre-id for **all** weather data. The **#** character is a multiple level wildcard for MQTT subscription, and can only be used at the end of a topic subscription.

By using **+** and **#** (the two defined wildcards in the MQTT[S] protocol), users can extend their subscription and receive all messages they need.

The purpose of additional topic hierarchy levels is to allow users to get the messages they need and to be more specific in their request.

4.1.3. Dos and don'ts when defining Earth system discipline topics

Define only a *small number* of additional levels

According to the MQTT[S] protocol specification, the accepted length of a given topic is more than 65kB. This means that the number of levels can be extremely large. However, as explained in <https://www.emqx.com/en/blog/advanced-features-of-mqtt-topics>:

Try not to use more topic levels “just because I can”. For example, `my-home/room1/data` is a better choice than `my/home/room1/data`.

By default, some MQTT brokers are configured to accept a maximum of 10 levels. This can be changed in the configuration of the broker, however, this limit shows that a usable, practical topic structure should not be too deep (i.e. with a large number of levels).

For WIS2, and considering the various Earth system disciplines, a limit of **4** sublevels seems appropriate.

Do not use the topic as a metadata record

When defining the topic, experts must focus on the needs of users. The purpose of the WIS2 Topic Hierarchy is **not** to describe as precisely as possible what data users will obtain if they decide to subscribe and then download. Rather, it is *only* to provide a filtering mechanism so that users will not be flooded by WIS2 Notification Messages that may not be useful for them.

In WIS2, all datasets **must** be described using the [WMO Core Metadata Profile version 2\(WCMP2\)](#). Users will be able to discover the data they need by searching the WIS2 catalogue using (via search engines, directly, or from portals and applications). The topic hierarchy information will be part of the metadata record for data which provides real-time notifications of publication.

*Do not allow locally defined sublevels outside the **experimental** topic

Each Earth system discipline provides an **experimental** topic. For example, for **weather**, the first additional levels in the domain topic hierarchy are:

Name	Description
advisories-warnings	Advisories and warnings
aviation	Aviation
prediction	Data sets produced by quantitative algorithms, such as numerical or statistical prediction models, describing the past, present and future meteorological states
space-based-observations	Space based observations
surface-based-observations	Surface based observations
experimental	Experimental topics

As the name suggests, **experimental** allows for defining additional topics for tests and experiments. This is not meant to be used for operational data exchange. It should only be used for testing purposes.

With the exclusion of the **experimental** topic level, the topic Hierarchy must be *fully* defined for each Earth system discipline.

In some situations, it might be tempting for a data producer to use additional topic levels to restrict even more the number of messages received by the users.

This is **forbidden**.

In each Earth system discipline community, all WIS Centres will be able to use the entire topic hierarchy of the domain if they provide data corresponding to each topic. A WIS Centre will not be allowed to add additional sublevels or undefined level within the Topic Hierarchy for its own needs.

Modification of the Topic Hierarchy will be possible by using the WMO fast-track approval process.

Consider users needs and prevent complex wildcard subscriptions

The purpose of the WIS2 Topic Hierarchy is to inform users about the availability of new data. In WIS2, obtaining data will start, in most cases, by configuring one or more subscription to topics, as defined in the associated WCMP2 discovery metadata records, so that users will receive notifications when new data is available.

The Topic Hierarchy should be defined so that users will not need to configure a very large number of different subscriptions to get the data they are interested in.

Each level in the WIS2 Topic Hierarchy should be seen as a "logical" group (as the Earth system disciplines **weather**, **ocean**... or like **synop** for **surface-based-observations**).

Then, and considering that wildcard subscription (using '+' and '#' as described above), are "expensive" to manage for the brokers.

For example, a topic hierarchy resulting in users subscribing to:

```
cache/a/wis2/+core/data/ocean/+some/+thing/+else/#
```

should be avoided. A subscription to the following topic:

```
cache/a/wis2/+core/data/ocean/some/thing/else/#
```

is much more effective for both the client and the broker side.

If it is likely that most users will use wildcards for particular topic levels, then, either removing that level altogether, or moving that level to the end of the topic hierarchy is also more efficient for clients and producers.

If most users end up subscribing to:

```
cache/a/wis2/+core/data/ocean/+thing/+/#
```

then, the Topic Hierarchy could be reconsidered, so that the above subscription can be replaced by:

```
cache/a/wis2/+core/data/ocean/thing/#
```

Reordering the levels of topics and potentially reducing the number of sublevels makes the topic hierarchy simpler and more efficient.

Facilitate client side filtering

Notification messages are small pieces of information. MQTT[S] broker and clients are able to handle a very large number of messages. In that sense, receiving, potentially, too many messages is not a problem. However, downloading data, depending on the size of the data might be slower and less efficient. If, for a particular dataset, the geometry information available in the notification message is not sufficient to allow client-side filtering before download, it is suggested to provide additional information in the `properties` object of the notification message so that users can decide *before* downloading if the data in this particular message is useful for them.

See [\[advertise_client_side_filters_for_data_subscriptions_in_wcmp2_and_wnm\]](#) for more information on client side filtering

Chapter 5. Recipes for Global Service operators