

OFFENBACH TRAINING WORKSHOP · October 2026

WIS2-in-a-box Overview

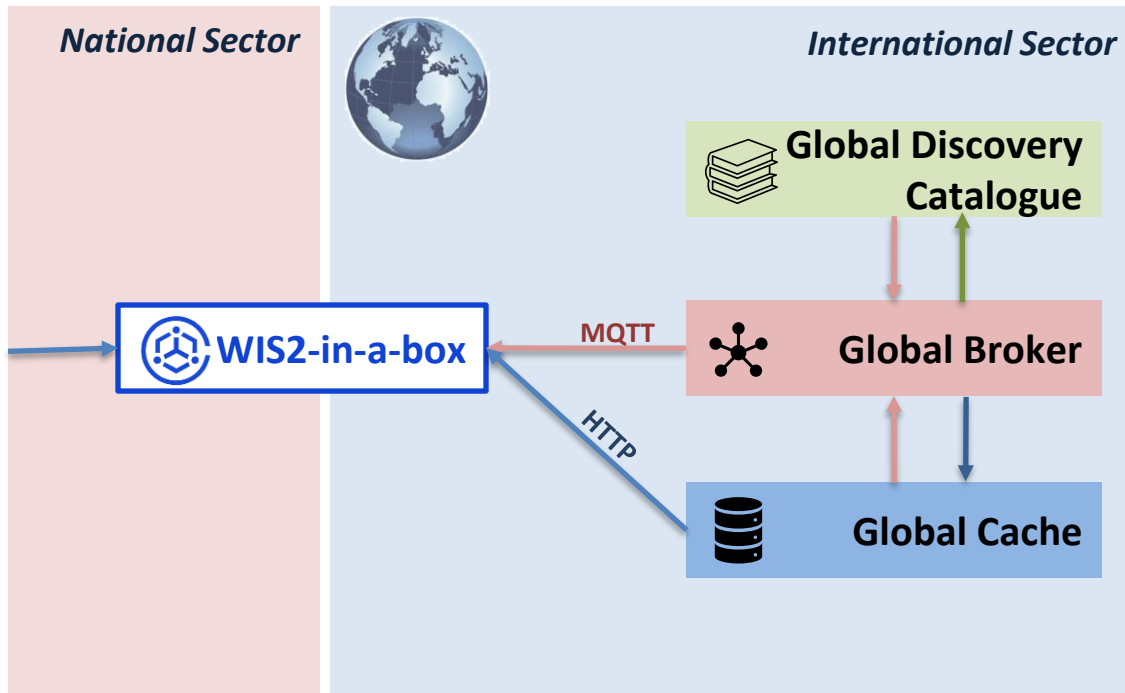
Maike Limper

WMO Secretariat · WIS Section · ESDP Department

What is WIS2-in-a-box ?

WIS2-in-a-box (wis2box) is an open-source software stack that can be installed on a server to setup a WIS2 Node

The wis2box-software includes all services required to run a WIS2 Node ... *and more !*



How to turn on wis2box-instance into a WIS2 Node:

- HTTP and MQTT traffic on the server hosting the wis2box-stack must be routed to the public internet
- MQTT-endpoint and centre-id to be shared with Global Brokers through the *WIS2 Node Registration process**

WIS2 in a box

- WIS2 in a box (wis2box) is a **Reference Implementation of a WIS2 Node**
- Developed by WMO together with Canada to **accelerate the WIS2 implementation**
- Designed to be **cost-effective** and **low-barrier** to operate*
- Developed as Docker Compose stack using existing **Free and Open Source** software and wis2box-specific components
 - wis2box source code at: <https://github.com/World-Meteorological-Organization/wis2box>
- Currently over 80 WMO-Members are using wis2box to share data on WIS2

*wis2box hosting requirements:

- minimum 2 vCPUs with 4GB Memory and 24GB of local storage
- requires Python, Docker and Docker Compose pre-installed
- HTTP and MQTT ports routed to a publicly accessible address
- See documentation at <https://docs.wis2box.wis.wmo.int>

WIS2 in a box is Free and Open

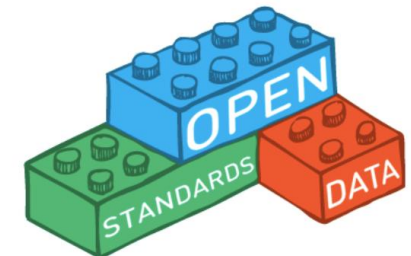
Free and Open Source Software



Open Standards



- MQTT
- GeoJSON
- OGC APIs



Docker and Docker Compose

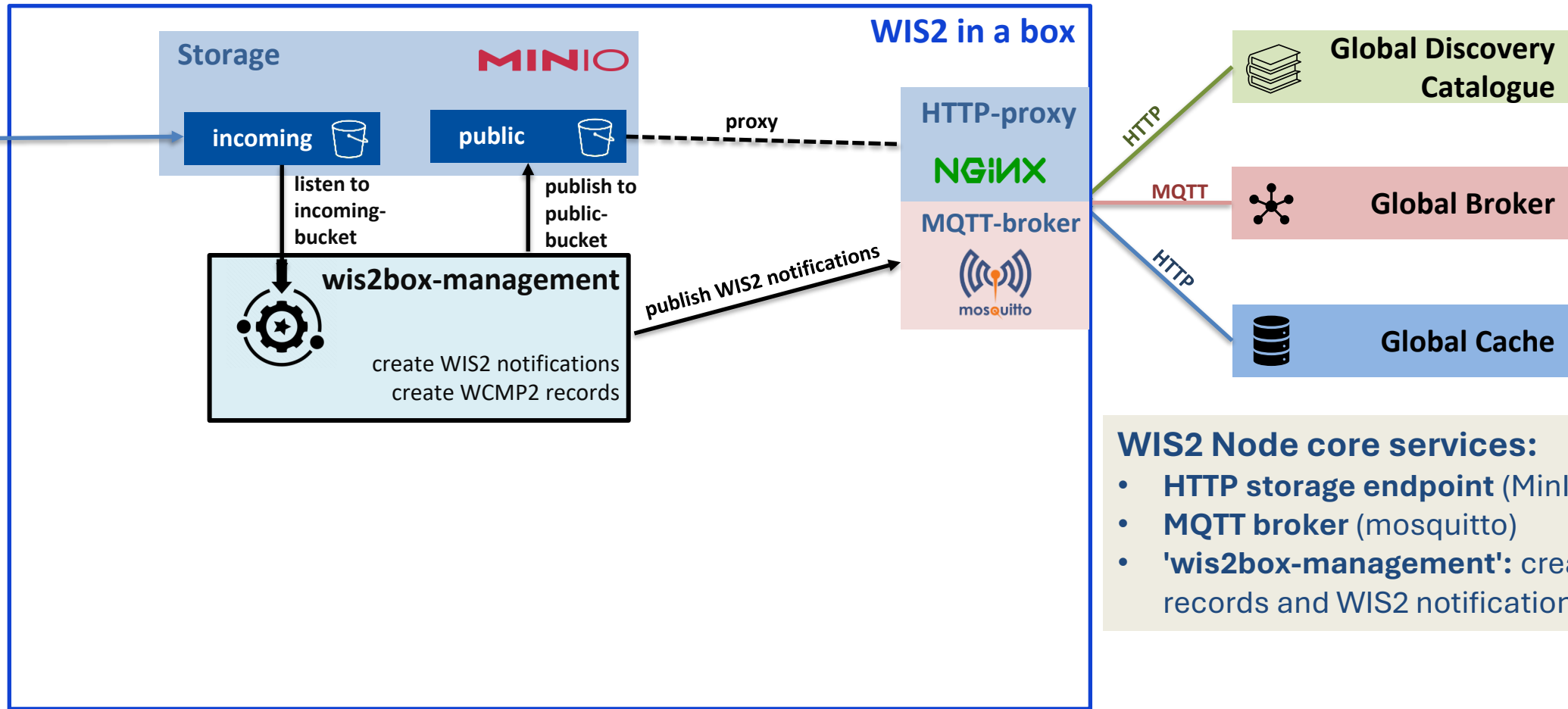
wis2box uses Docker and Docker Compose to define a set of services

- you do not need to be familiar with Docker to run wis2box
- *wis2box-setup-*.zip* contains all files needed to start the wis2box-stack
- pre-built Docker images are used to create containers providing a specific service

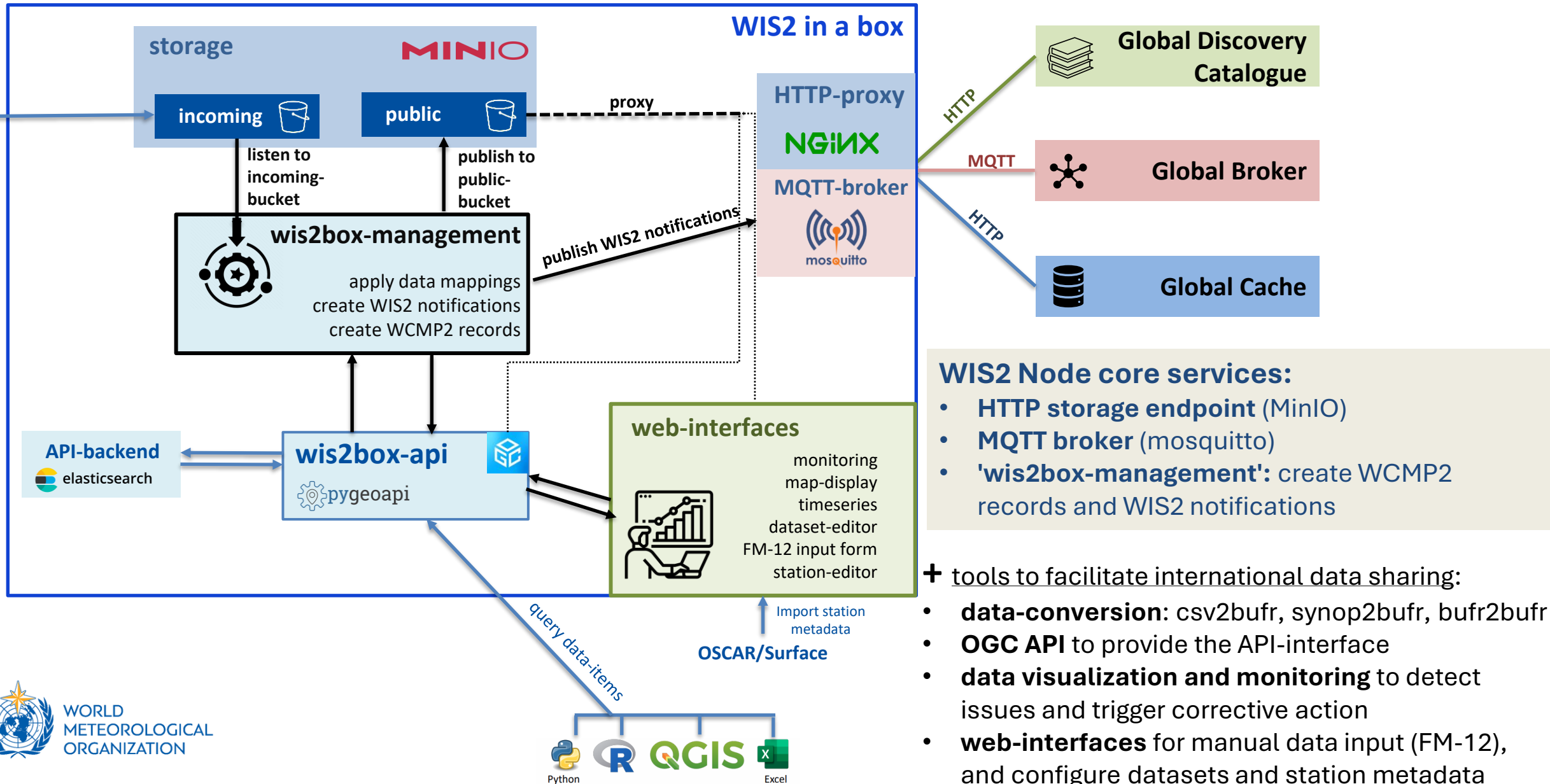
Python script “wis2box-ctl.py” provide wrapper for docker compose commands



WIS2 in a box core services



WIS2 in a box services



How to setup a WIS2 Node using wis2box

Step 1: Find a host that can be routed to be public internet

Recommended options:

- Virtual Machine in your local data-centre (same place as other web services)
- Virtual Machine provided by a 3rd party (a.k.a. “Cloud”)



Step 2: Setup your host according to wis2box requirements:

<https://docs.wis2box.wis.wmo.int/en/latest/user/getting-started.html>

The following commands be used to setup the required software on Ubuntu (22.04 LTS) systems:

Note

Execute the below commands one by one, and do not copy / paste the entire block

```
sudo mkdir -m 0755 -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor --yes -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
sudo echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list
sudo apt-get -y update
sudo apt-get install -y docker-ce docker-compose-plugin unzip python3-pip
sudo pip3 install pip --upgrade
sudo pip3 install pyopenssl --upgrade
sudo pip3 install urllib3==1.26.0
```

Step 3: Install and setup wis2box on your host:

<https://docs.wis2box.wis.wmo.int/en/latest/user/setup.html>

wis2box workflow

A fresh installation of wis2box has all required services to start functioning as a WIS2 Node

... but before your wis2box can publish anything you must define datasets :

- Using the dataset-editor in the wis2box-webapp, or
- From the command line in the wis2box-management container: `wis2box dataset publish <mcf-file>`

Datasets drive the wis2box workflow for publishing WIS2 notifications:

- A Dataset is described by **Discovery Metadata** that is published as a WCMP2-record
- A Dataset **connects data and metadata** using the **metadata identifier**
- A Dataset **defines the topic hierarchy** to publish the WIS2 data notification
- A Dataset in wis2box contains **data plugins** used to **transform** and **publish** the data

How to configure datasets in wis2box: using the dataset editor



WIS2 in a box

Dataset Editor Form

Please choose a dataset

Dataset

Select existing discovery metadata file or
create a new file.

Create New...

New dataset: Initial Information ?

Centre ID

int-wmo-example

Template

other

climate/surface-based-observations/daily

ocean/surface-based-observations/drifting-buoys

weather/surface-based-observations/ship

weather/surface-based-observations/synop

weather/surface-based-observations/temp

other

Centre ID:

Select from dropdown or type any valid string
defines the metadata-id and centre-id in topic

Template:

Select option from dropdown

Select **template** to initialize the dataset with a fixed topic
and pre-defined values for keywords and data mappings

Select **'other'** for any types of data you wish to publish,
and you will select a topic based on the WIS2 Topic
Hierarchy

How to configure datasets in wis2box: using the dataset editor

Dataset Editor Form

Please choose a datasetDataset

Dataset loaded successfully.

Metadata Editor

Dataset Identification

Title

Hourly synoptic observations from fixed-land stations (SYNOP) (br-inmet)

Description

Observation data from automatic weather stations

Identifier

urn:wmo.md:br-inmet:surface-based-observations.synop

Centre ID

br-inmet

WMO Data Policy

core

Topic Hierarchy

br-inmet/data/core/weather/surface-based-observations/synop

Earth System Disciplines

Weather

Keywords (3 minimum)

observations

temperature

visibility

precipitation

pressure

clouds

snow depth

evaporation

radiation

wind

total sunshine

humidity

Temporal Properties

2024-06-05

End Date in UTC

☒ Dataset ongoing

Resolution

1

Unit

hour(s)

Spatial Properties

Choose an automatic bounding box (optional)

Brazil

Your country may not have an automatic bounding box

North Latitude

5.24448639!

West Longitude

-73.9872354

South Latitude

-33.7683777

East Longitude

-34.7299934



Step 1. Define metadata and validate form

Step 2. Define data plugins

Step 3. Provide Authentication Token (created by wis2box-admin)

Step 4. Submit the dataset for publication

Contact Information of the Data Provider

Organization Name

WMO

URL

https://wmo.int

Country

Switzerland

Email

wis2-support@wmo.int

Phone number (optional)

RESET FORM

VALIDATE FORM

Dataset Mappings Editor

Plugins in use	File extension	File pattern		
BUFR data converted to BUFR	bin	^.*\.bin\$	UPDATE	DELETE
FM-12 data converted to BUFR	txt	^.*_(\d{4})(\d{2}).*\.txt\$	UPDATE	DELETE
BUFR data converted to BUFR	b	^.*\.b\$	UPDATE	DELETE
CSV data converted to BUFR	csv	^.*\.csv\$	UPDATE	DELETE
BUFR data converted to GeoJSON	bufr4	^WIGOS_(\d+\d+-\d+-\d+)_.*\.bufr4\$	UPDATE	DELETE

A ADD A PLUGIN

Authentication Token

wis2box auth token for 'processes/wis2box'

EXPORT AS JSON

SUBMIT

METEOROLOGICAL ORGANIZATION

Datasets in wis2box: data plugins



Data Mappings map a specific **dataset** to a set of **data plugins**

Data plugins use an abstract model/approach to enable extensibility and reuse

A **data plugin** defines the actions taken to **transform** and **publish** the data

See github.com/wmo-im/wis2box/tree/main/wis2box-management/wis2box/data

NOTE:

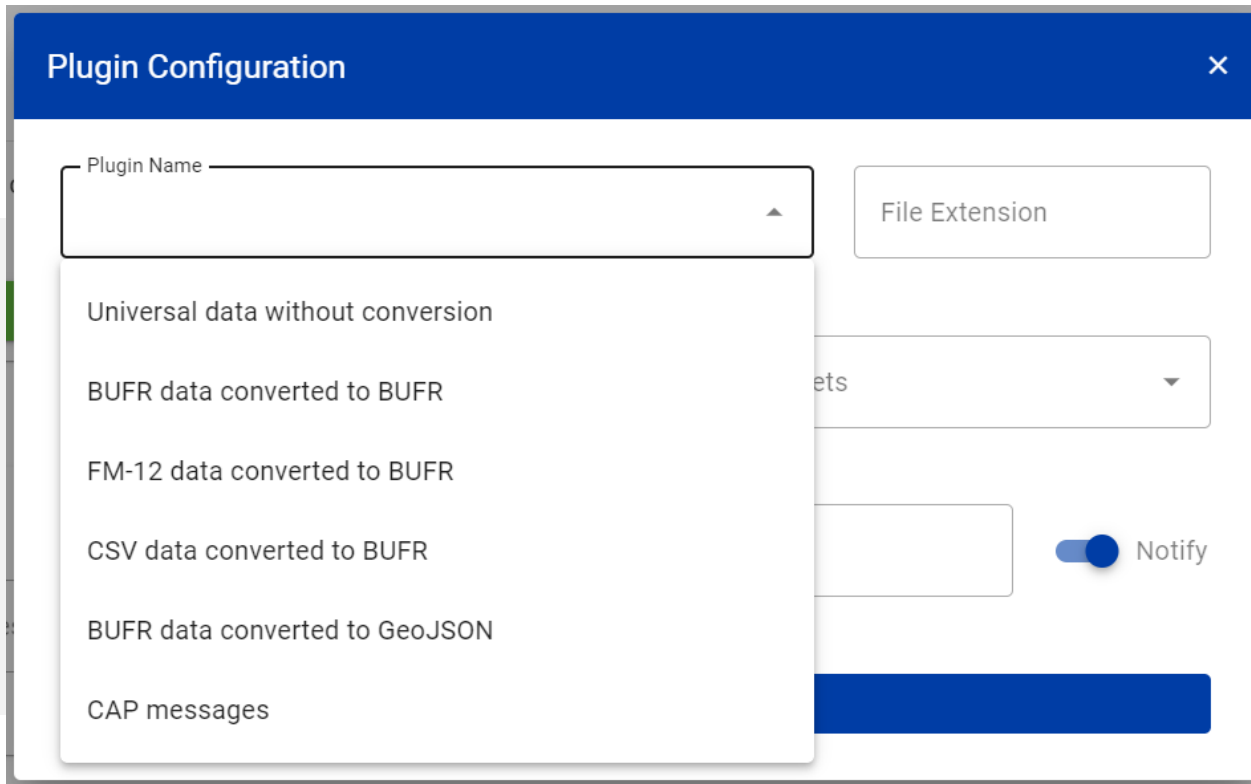
Preconfigured data-plugins in data mappings for template="weather/surface-based-observations/synop" in dataset-editor, can be adjusted as required

Plugins in use	File extension	File pattern
BUFR data converted to BUFR	bin	^.*\.bin\$
FM-12 data converted to BUFR	txt	^.*_(\d{4})(\d{2}).*.txt\$
BUFR data converted to BUFR	b	^.*\.b\$
CSV data converted to BUFR	csv	^.*\.csv\$
BUFR data converted to GeoJSON	bufr4	^WIGOS_(\d-\d+-\d+-\d+)_.*\.bufr4\$

wis2box data plugins

wis2box contains the following built-in data plugins:

- wis2box.data.universal.UniversalData
- wis2box.data.cap_message.CAPMessageData
- wis2box.data.bufr4.ObservationDataBUFR
- wis2box.data.synop2bufr.ObservationDataSYNOP2BUFR
- wis2box.data.csv2bufr.ObservationDataCSV2BUFR
- wis2box.data.bufr2geojson.ObservationDataBUFR2GeoJSON



The screenshot shows a 'Plugin Configuration' dialog box with a blue header and a close button (X) in the top right corner. Inside the dialog, there is a 'Plugin Name' dropdown menu that is open, displaying a list of built-in plugins: 'Universal data without conversion', 'BUFR data converted to BUFR', 'FM-12 data converted to BUFR', 'CSV data converted to BUFR', 'BUFR data converted to GeoJSON', and 'CAP messages'. To the right of the dropdown is a 'File Extension' input field. Below the dropdown is another dropdown menu with the text 'ets' and a downward arrow. To the right of this is a 'Notify' toggle switch, which is currently turned on. At the bottom of the dialog is a solid blue button.

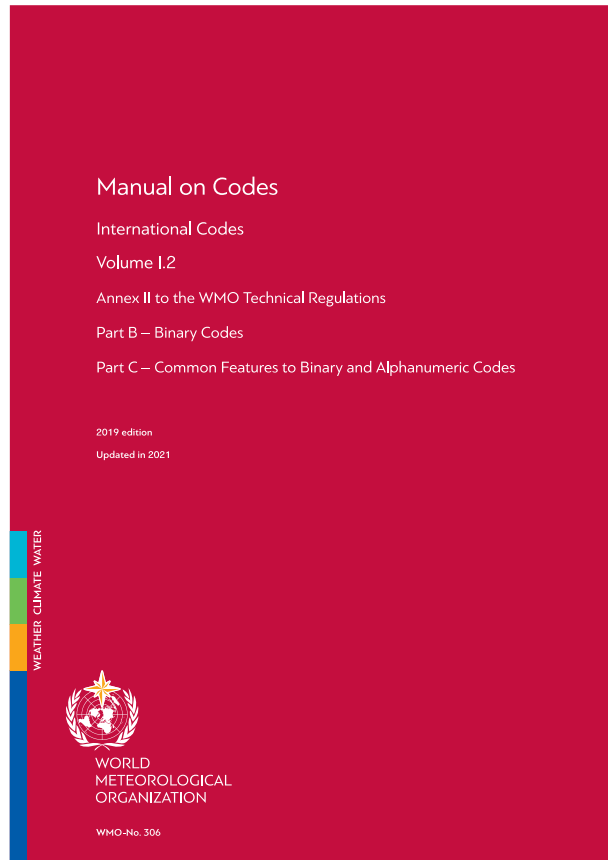
Developers are encouraged to contribute new data plugins to wis2box!

BUFR ?

BUFR (Binary Universal Form for the Representation of meteorological data)

Note: WIS2 regulations do not specify data format for the exchanged data within the notifications !

WMO Manual on Codes defines how internationally exchanged data for meteorology is encoded



How BUFR Works:

BUFR message consists of a **header** and a **body containing data and metadata**.
BUFR descriptors define how data is structured and encoded within a BUFR message
Compression for efficient storage to reduce transmission times.

BUFR Table System

BUFR descriptors are stored in **BUFR Tables** (Tables A, B, C, and D).

- **Table B** contains all element descriptors with their definitions.
- **Table D** defines **combinations of descriptors** into predefined sequences

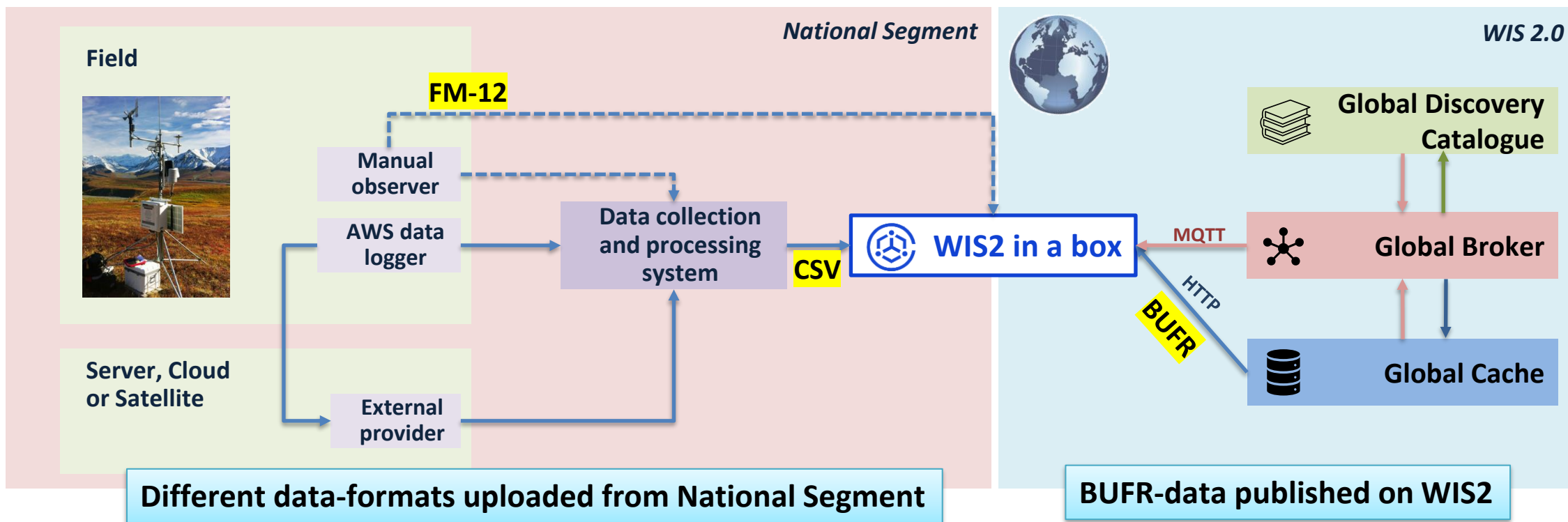
BUFR tables are part of the **WMO Manual on Codes**

<https://library.wmo.int/idurl/4/35625> new version is released **every six months**

BUFR tools in wis2box

wis2box data plugins use the BUFR tools to enable data transformation prior to publication

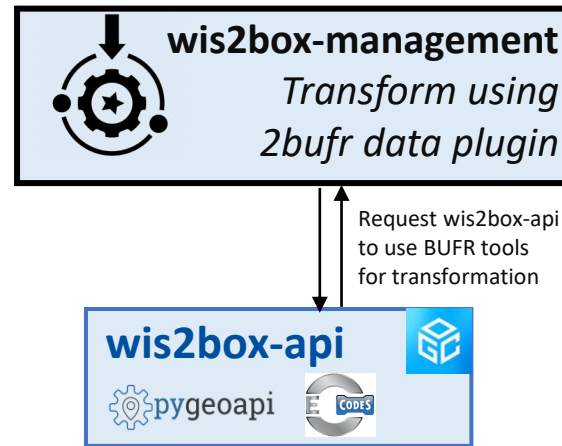
- **csv2bufr** enables any system to prepare a data-extract without needing local BUFR conversion tools
- **synop2bufr** enables direct publication of FM-12 synop reports from manual observers
- **bufr2bufr** splits bulletins and ensure data has a valid WIGOS-station-identifier



BUFR tools in wis2box



‘wis2box-api’ container is based on a Docker image containing [ecCodes](#) software library ... the container also includes several BUFR-conversion tools that work with ecCodes



BUFR tools used within wis2box-api available to be used standalone:

<https://github.com/World-Meteorological-Organization/csv2bufr>

<https://github.com/World-Meteorological-Organization/synop2bufr>

<https://github.com/World-Meteorological-Organization/bufr2geojson>

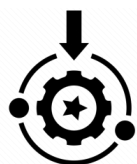
wis2box data plugins: synop2bufr

File containing SYNOP
messages (FM-12)

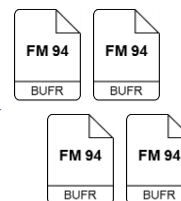


Station list

synop2bufr



One or more
BUFR files



bufr2geojson (optional)



GeoJSON

Plugin Name

Universal data without conversion

BUFR data converted to BUFR

FM-12 data converted to BUFR

CSV data converted to BUFR

BUFR data converted to GeoJSON

CAP messages

plugins:

txt:

- plugin: wis2box.data.synop2bufr.ObservationDataSYNOP2BUFR
- notify: true
- file-pattern: '^*(\d{4})(\d{2}).*\.txt\$'

bufr4:

- plugin: wis2box.data.bufr2geojson.ObservationDataBUFR2GeoJSON
- file-pattern: '^WIGOS_(\d-\d+-(\d+-(\w+))_.*\bufr4\$'

wis2box data plugins: csv2bufr

Tabulated CSV data from
observing station, including
location



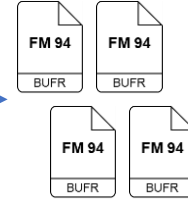
Station list

csv2bufr



mapping template

One or more
BUFR files



bufr2geojson (optional)



GeoJSON

Plugin Name

Universal data without conversion

BUFR data converted to BUFR

FM-12 data converted to BUFR

CSV data converted to BUFR

BUFR data converted to GeoJSON

CAP messages

plugins:

csv:

- plugin: wis2box.data.csv2bufr.ObservationDataCSV2BUFR
- template: aws-template.json
- notify: true
- file-pattern: '^.*\.csv\$'

bufr4:

- plugin: wis2box.data.bufr2geojson.ObservationDataBUFR2GeoJSON
- file-pattern: '^WIGOS_(\d-\d+-(\d+-(\w+))_.*\.bufr4\$'

wis2box data plugins: bufr2bufr

File containing one or more
BUFR subsets

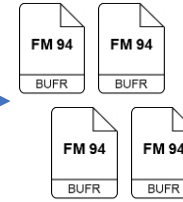
BUFR processing
(extraction of subsets)

One or more
BUFR files

bufr2geojson (optional)



Station list



GeoJSON

Plugin Name

- Universal data without conversion
- BUFR data converted to BUFR**
- FM-12 data converted to BUFR
- CSV data converted to BUFR
- BUFR data converted to GeoJSON
- CAP messages

plugins:

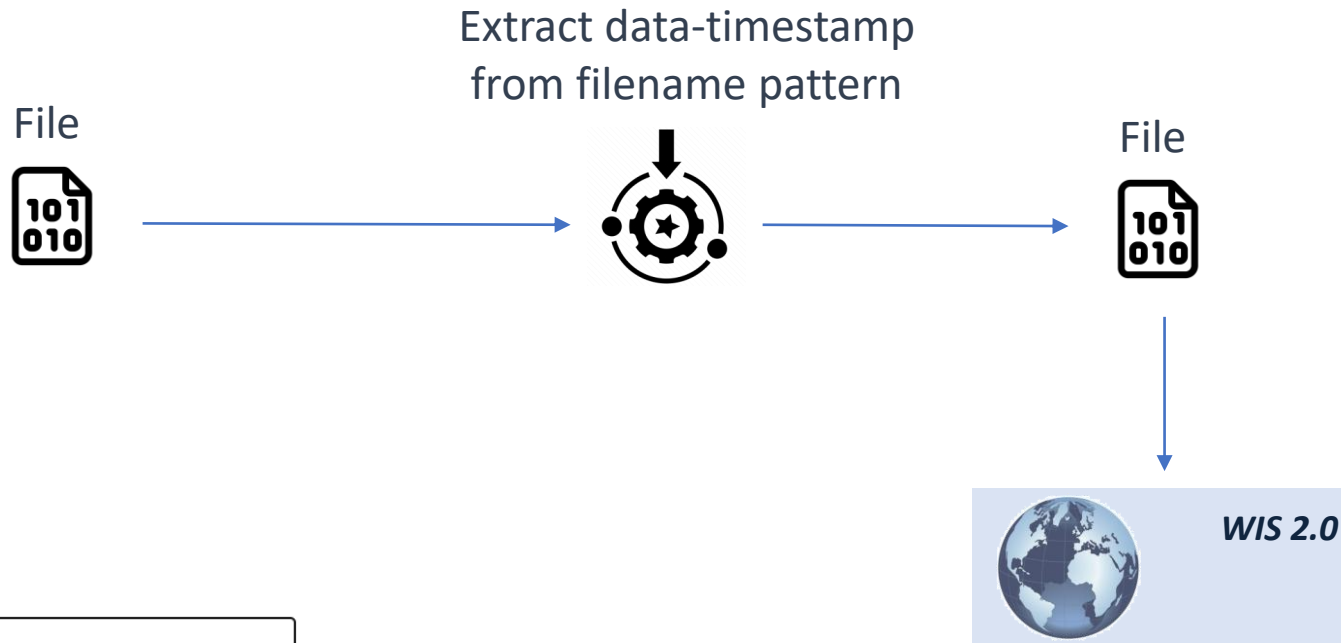
bin:

- **plugin:** wis2box.data.bufr4.ObservationDataBUFR
- notify:** true
- file-pattern:** '^.*\.bin\$'

bufr4:

- **plugin:** wis2box.data.bufr2geojson.ObservationDataBUFR2GeoJSON
- file-pattern:** '^WIGOS_(\d-\d+-\d+-\d+)_.*\.bufr4\$'

wis2box data plugins: universal/passthrough



Plugin Name

Universal data without conversion

BUFR data converted to BUFR

FM-12 data converted to BUFR

CSV data converted to BUFR

BUFR data converted to GeoJSON

CAP messages

plugins:

grib2:

- plugin: **wis2box.data.universal.UniversalData**

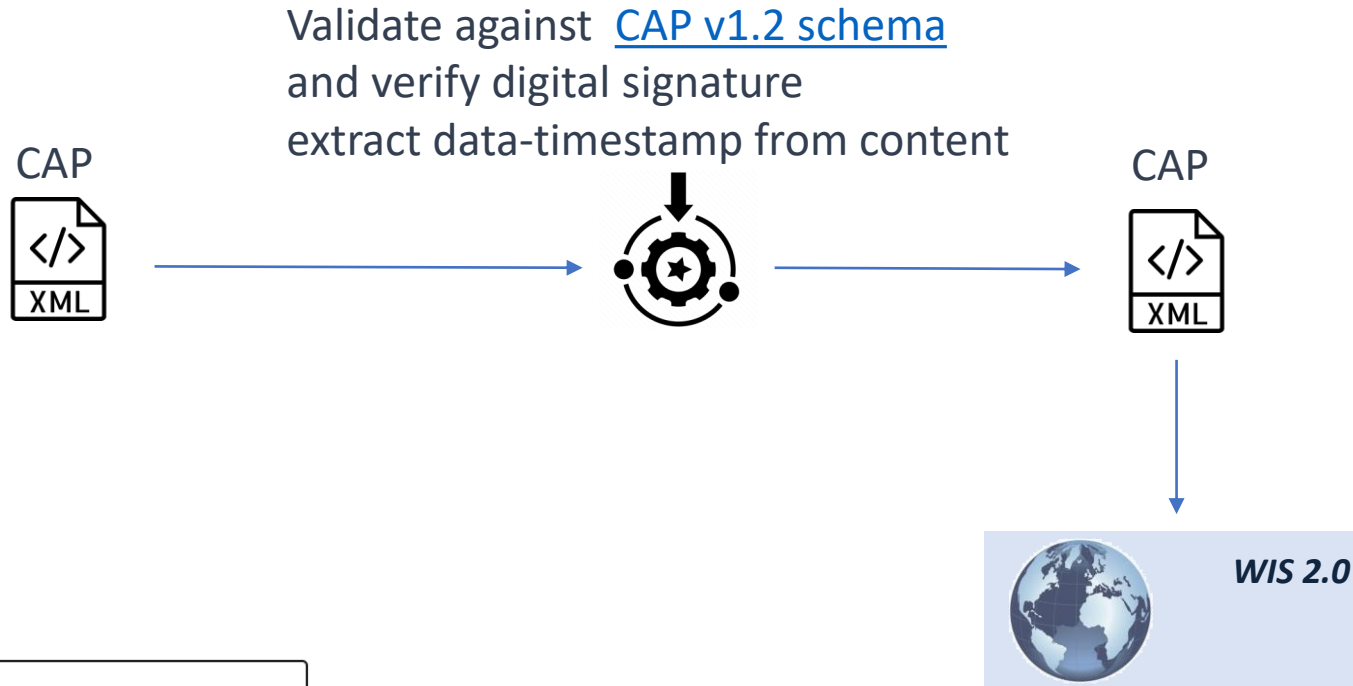
notify: true

buckets:

- **\${WIS2BOX_STORAGE_INCOMING}**

file-pattern: '^.*_(\d{8})\d{2}.*\.grib2\$'

wis2box data plugins: Common Alerting Protocol (CAP)



Plugin Name

- Universal data without conversion
- BUFR data converted to BUFR
- FM-12 data converted to BUFR
- CSV data converted to BUFR
- BUFR data converted to GeoJSON
- CAP messages**

plugins:

xml:

- plugin: `wis2box.data.cap_message.CAPMessageData`
- notify: true
- buckets:
 - `${WIS2BOX_STORAGE_INCOMING}`
- file-pattern: `'^.*\.xml$'`

CSV-to-BUFR introduction

wis2box contains a built-in data-plugin for “CSV data converted to BUFR”

- Enables ingesting data in CSV format while publishing WIS2 notifications containing BUFR data
- Uses a mapping template to map input columns to BUFR codes

csv2bufr-transformation is provided by the ‘csv2bufr’ module that can also be used outside of the wis2box <https://github.com/world-meteorological-organization/csv2bufr>

Plugin Configuration

Plugin Name
CSV data converted to BUFR

File Extension
CSV

Template
AWS

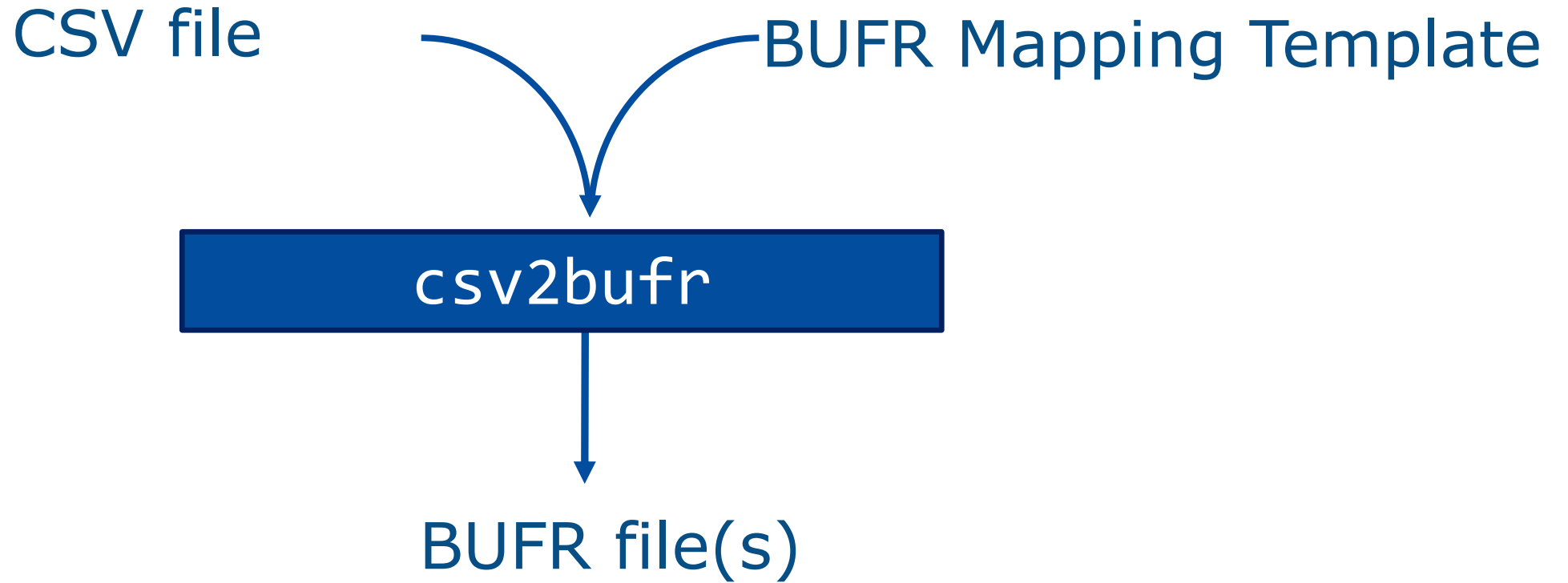
WIS2 Buckets
Incoming

File Pattern (Regex)
.*\.csv\$

☒ Notify

SAVE

CSV-to-BUFR key ingredients



BUFR mapping template

Template: JSON file defining mapping between columns in the input CSV data to codes encoded in the output BUFR data

... using the human-readable eccodes_key rather than the 6 digit BUFR FXXYYY code

```
"data": [
```

```
....
```

```
{ "eccodes_key": "#1#nonCoordinatePressure", "value": "data:station_pressure", "valid_min": "const:50000", "valid_max": "const:150000"},  
{ "eccodes_key": "#1#pressureReducedToMeanSeaLevel", "value": "data:msl_pressure", "valid_min": "const:50000", "valid_max": "const:150000"},  
{ "eccodes_key": "#1#nonCoordinateGeopotentialHeight", "value": "data:geopotential_height", "valid_min": "const:-1000", "valid_max": "const:130071"},  
{ "eccodes_key": "#1#heightOfSensorAboveLocalGroundOrDeckOfMarinePlatform", "value": "data:thermometer_height", "valid_min": "const:0", "valid_max": "const:655.35"},  
{ "eccodes_key": "#1#airTemperature", "value": "data:air_temperature", "valid_min": "const:193.15", "valid_max": "const:333.15"},  
... ]
```

Data CSV file

BUFR Mapping template

csv2bufr

BUFR file(s)

BUFR mapping template

valid_min: Minimum tolerable value

valid_max: Maximum tolerable value

csv2bufr uses 'valid_min' and 'valid_max' to catch input errors: invalid values will not be encoded in the BUFR

```
"data": [
```

```
....
```

```
{ "eccodes_key": "#1#nonCoordinatePressure", "value": "data:station_pressure", "valid_min": "const:50000", "valid_max": "const:150000"},  
{ "eccodes_key": "#1#pressureReducedToMeanSeaLevel", "value": "data:msl_pressure", "valid_min": "const:50000", "valid_max": "const:150000"},  
{ "eccodes_key": "#1#nonCoordinateGeopotentialHeight", "value": "data:geopotential_height", "valid_min": "const:-1000", "valid_max": "const:130071"},  
{ "eccodes_key": "#1#heightOfSensorAboveLocalGroundOrDeckOfMarinePlatform", "value": "data:thermometer_height", "valid_min": "const:0", "valid_max": "const:655.35"},  
{ "eccodes_key": "#1#airTemperature", "value": "data:air_temperature", "valid_min": "const:193.15", "valid_max": "const:333.15"},  
...]
```

Data CSV file

*air_temperature
value=29.5*

BUFR Mapping template

csv2bufr

*WARNING: Value (29.5) out of valid range
(193.15 - 333.15); Element set to missing*

BUFR file(s)

The AWS template

AWS Template:

Standardized CSV format to ingest data from Automatic Weather Stations in support of GBON reporting requirements

The format is intended for use with automatic weather stations reporting a minimum number of parameters, including pressure, air temperature and humidity, wind speed and direction and precipitation

Based on SI units, with no unit prefixes, e.g. Pa rather than hPa, Kelvin rather than degrees Celsius etc

Template = AWS in the Plugin Configuration
refers to the '**aws-template.json**' as defined in
<https://github.com/wmo-im/csv2bufr-templates>

Plugin Configuration

Plugin Name: CSV data converted to BUFR

File Extension: CSV

Template: AWS

WIS2 Buckets: Incoming

File Pattern (Regex): *.*.csv\$

Notify: ☒

SAVE

The DAYCLI template

DAYCLI Template:

Standardized CSV format for converting daily climate data to BUFR sequence 307075

The format is intended for use with Climate Data Management Systems to publish data on WIS2

Requirement for daily climate observations to monitor extremes

Daily observations of:

- Minimum, maximum and average temperature over 24 hours period
- Total accumulated precipitation over 24 hours period
- Total snow depth at time of observation
- Depth of fresh snow over 24 hours period

Recognition of different reporting practices by Members, explicit reporting of 24 hours period used

Additional metadata: method of calculating average temperature; sensor and station heights; exposure and measurement quality classification

Template = AWS in the Plugin Configuration
refers to the '**daycli-template.json**' as defined in
<https://github.com/wmo-im/csv2bufr-templates>

Plugin Configuration

Plugin Name
CSV data converted to BUFR

File Extension
CSV

Template
DayCLI

WIS2 Buckets
Incoming

File Pattern (Regex)
.*\.csv\$

☒ Notify

SAVE

SYNOP-to-BUFR

SYNOP FM-12 text file

Station list CSV file

synop2bufr

synop2bufr

<https://github.com/wmo-im/synop2bufr>

csv2bufr

pymetdecoder

csv2bufr

<https://github.com/wmo-im/csv2bufr>

pymetdecoder

<https://github.com/antarctica/pymetdecoder>

UK Research and Innovation (UKRI), 2021, British Antarctic Survey

BUFR file(s)

SYNOP-to-BUFR

AAXX = data from fixed
land station

FM-12 SYNOP reports

SMR001 YRBK 211200 **YY = 21 of the month**
GG = 12 hour of observation
iw = 1 Wind speed obtained from anemometer in m/s
AAXX 21121
15015 02999 02501 10103 21090 39765 42952 57020 60001 333 4/000 55310
0//// **TSI**
22591 3//// 60007 91003 91104=
15020 02997 23104 10130 21075 30177 40377 58020 60001 81041 333 4/000
55310 **TSI**
0//// 22547 3//// 60007 91008 91111=

Station list metadata

station_name	wigos_station_identifier	traditional_station_identifier	facility_type	latitude	longitude	elevation
OCNA SUGATAG	0-20000-0-15015	15015	Land (fixed)	47.7770616	23.9404603	503
BOTOSANI	0-20000-0-15020	15020	Land (fixed)	47.7356532	26.6455502	161
IASI	0-20000-0-15090	15090	Land (fixed)	47.1633333	27.6272222	74.29
CEAHLAU TOACA	0-20000-0-15108	15108	Land (fixed)	46.97751	25.94994	1897
CLUJ-NAPOCA	0-20000-0-15120	15120	Land (fixed)	46.7777705	23.5713053	410
BACAU	0-20000-0-15150	15150	Land (fixed)	46.5577778	26.8966667	174
MIERCUREA CIUC	0-20000-0-15170	15170	Land (fixed)	46.3713167	25.7726167	661
ARAD	0-20000-0-15200	15200	Land (fixed)	46.1335164	21.3536215	116.59
DEVA	0-20000-0-15230	15230	Land (fixed)	45.864923	22.8988062	240
SIBIU	0-20000-0-15260	15260	Land (fixed)	45.79018	24.036245	450
VARFU OMU	0-20000-0-15280	15280	Land (fixed)	45.4457928	25.456691	2504
CARANSEBES	0-20000-0-15292	15292	Land (fixed)	45.41667	22.22917	241
GALATI	0-20000-0-15310	15310	Land (fixed)	45.4729181	28.0323011	69
TULCEA	0-20000-0-15335	15335	Land (fixed)	45.1905065	28.8241608	4.36
RAMNICU VALCEA	0-20000-0-15346	15346	Land (fixed)	45.0888211	24.3628139	237
BUZAU	0-20000-0-15350	15350	Land (fixed)	45.1326633	26.8517319	97
SULINA	0-20000-0-15360	15360	Land (fixed)	45.1623111	29.7268286	69
DROBETA-TURNU SEVERIN	0-20000-0-15410	15410	Land (fixed)	44.6264587	22.6260737	77
BUCURESTI BANEASA	0-20000-0-15420	15420	Land (fixed)	44.510433	26.0781904	90

Station metadata used in transformation to BUFR

FM-12 to BUFR (synop2bufr):

- **Add the WIGOS-station-ID**, derived by matching with traditional station identifier in FM-12 input
- **Add the latitude, longitude**, altitude and barometer height above sea-level in the BUFR content

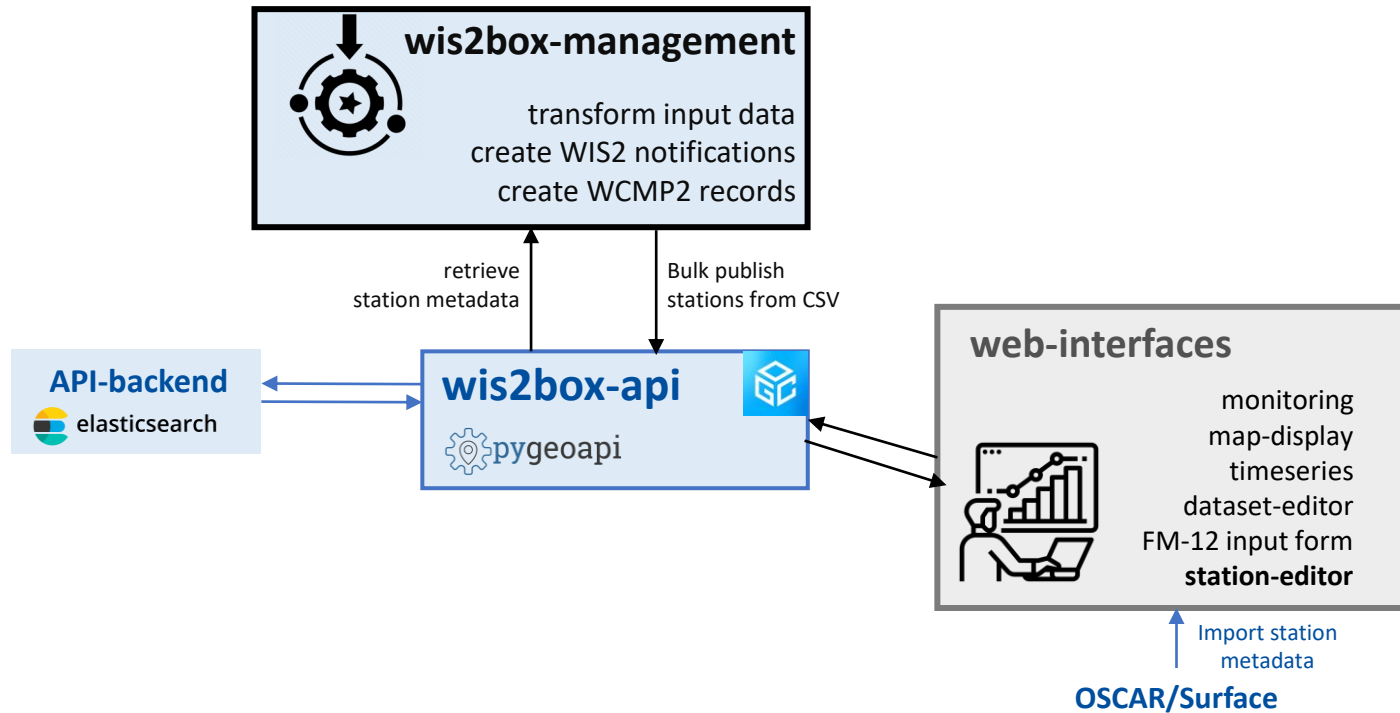
CSV to BUFR (csv2bufr):

- Check **WIGOS-station-ID** in input data is in station-list for associated topic
- For fixed land stations: check input-data-location is less than 1km from station metadata location

BUFR to BUFR (bufr2bufr):

- Splits by bulletin and **check WIGOS-station-ID** in station-list for associated topic
- If WIGOS-station-ID missing add it (by matching with traditional station identifier)
- if the location or station elevation are missing in the input BUFR use values from station metadata

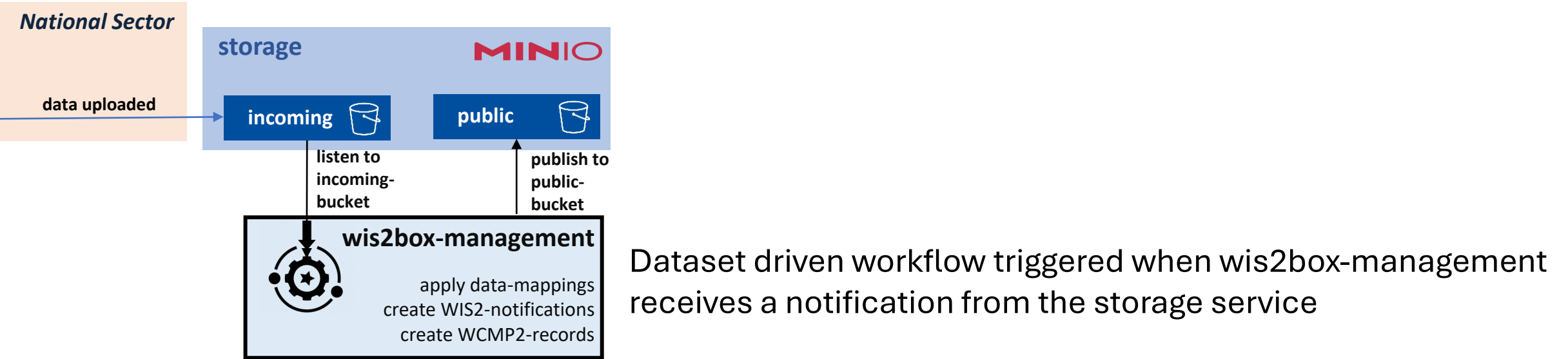
Station metadata in the wis2box



“stations” collection in wis2box-api can be populated in multiple ways:

- Provide station list as CSV and use command line inside wis2box-management container
- Interactively using the station editor inside the wis2box-webapp

wis2box data ingest workflow



wis2box-management **matches incoming data** with **datasets** based on file-path
data plugins are applied to **transform & publish the data**

How to upload data to wis2box



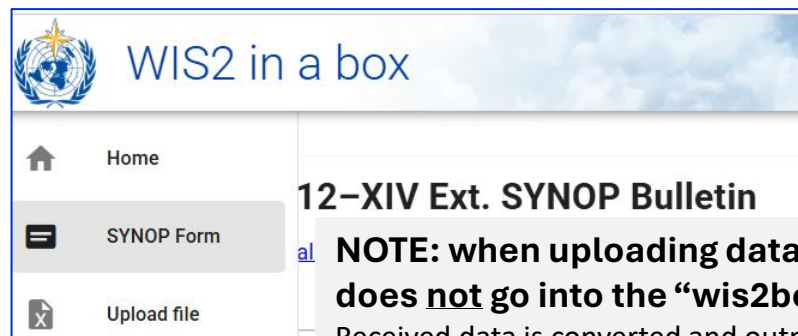
Data can be uploaded to wis2box-incoming in different ways
Optimal solution will depend on systems used in national sector

Manual upload methods:

- [Interfaces in wis2box-webapp](#): manual submission using FM-12 form or manual file upload interface
- [Using MinIO UI](#): manually create path and upload data, useful for testing data-workflow

Automating data upload (recommended!):

- [Scripts](#): write code to upload data using MinIO client software
- [Using SFTP protocol](#): forward data to MinIO SFTP-endpoint



NOTE: when uploading data in wis2box-webapp, received data does not go into the “wis2box-incoming” bucket
Received data is converted and output is stored in “wis2box-public” bucket directly

The wis2box homepage

The wis2box home-page displays the datasets configured and offers some options to view SYNOP / TEMP data

- links to “metadata” and “messages” provided for all datasets

Welcome to WIS 2.0 in a box!

Filter datasets by



3Hourly synoptic observations from fixed-land stations (SYNOP) (pg-pngnws)

Topic: origin/a/wis2/pg-pngnws/data/core/weather/surface-based-observations/synop
Metadata Identifier: urn:wmo:md:pg-pngnws:synop-3hourly

EXPLORE ⓘ OBSERVATIONS 📉 METADATA 📄 MESSAGES 📧



CAP Alerts published by pg-pngnws

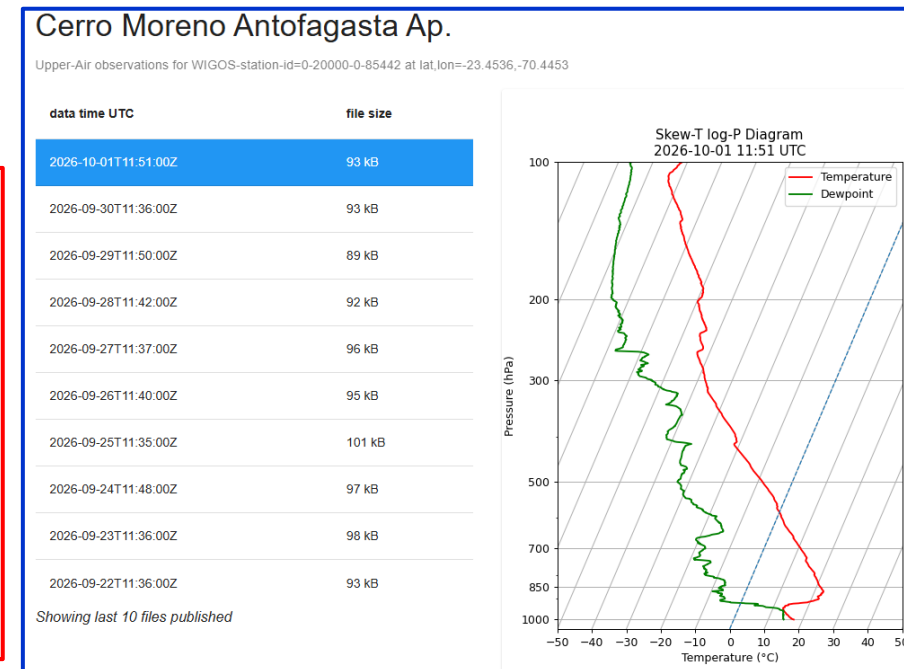
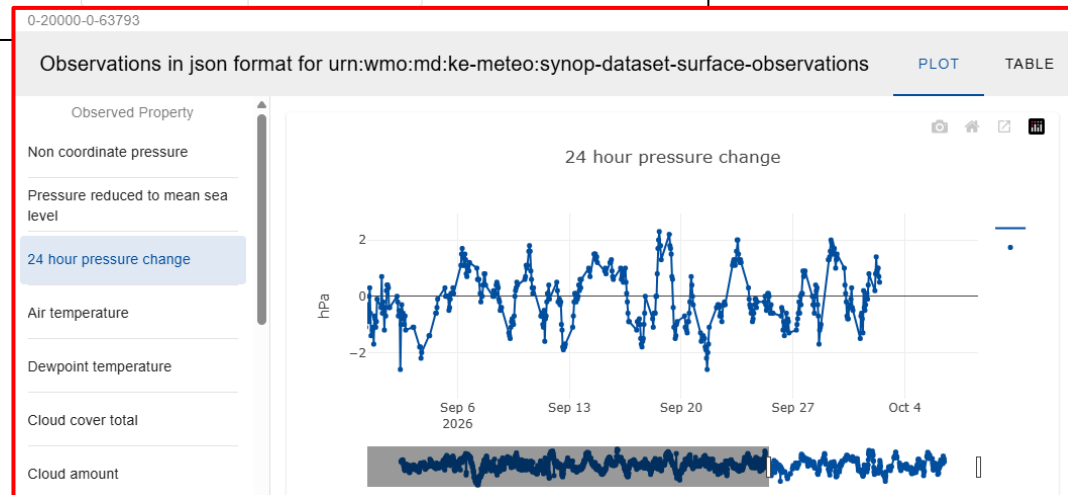
Topic: origin/a/wis2/pg-pngnws/data/core/weather/advisories-warnings
Metadata Identifier: urn:wmo:md:pg-pngnws:cap

METADATA 📄 MESSAGES 📧

SYNOP: The links for “explore” and “observations” are **only available when bufr2geojson is enabled** as the viewer relies on geoJSON stored in the API-backend

TEMP: From wis2box release 1.2.0 there is the option to “explore” this data, **this view does not rely on geoJSON stored in the backend as it decodes the published BUFR on the fly**

It is not advised to enable bufr2geojson on TEMP-data in wis2box



WIS2 in a box Overview

wis2box is a Free and Open Source Reference Implementation of a WIS2 Node

- Developers can freely use components of wis2box to adapt existing systems to be WIS 2.0 compliant
- Documentation: <https://docs.wis2box.wis.wmo.int>
- Source code: <https://github.com/World-Meteorological-Organization/wis2box>
- Includes **core services** needed to publish data on WIS2 and **optional services** to support international data exchange as per WMO standards
- ***Feedback by the community is appreciated to help improve wis2box !***

Datasets in wis2box use data plugins that can (optionally) transform the input data before publication

- **One wis2box-instance can host multiple WIS2 Nodes**, by configuring datasets with different centre-ids

wis2box is software not hardware, it requires a host on a (virtual) server

- Minimum system requirements: 2 vCPUs, 4GB Memory, 24GB storage
- Requires Python, Docker and Docker Compose to be pre-installed
- To function as a WIS2 Node, the wis2box-host needs HTTP and MQTT traffic routed to public IP

Thank you.



WORLD
METEOROLOGICAL
ORGANIZATION

wmo.int