

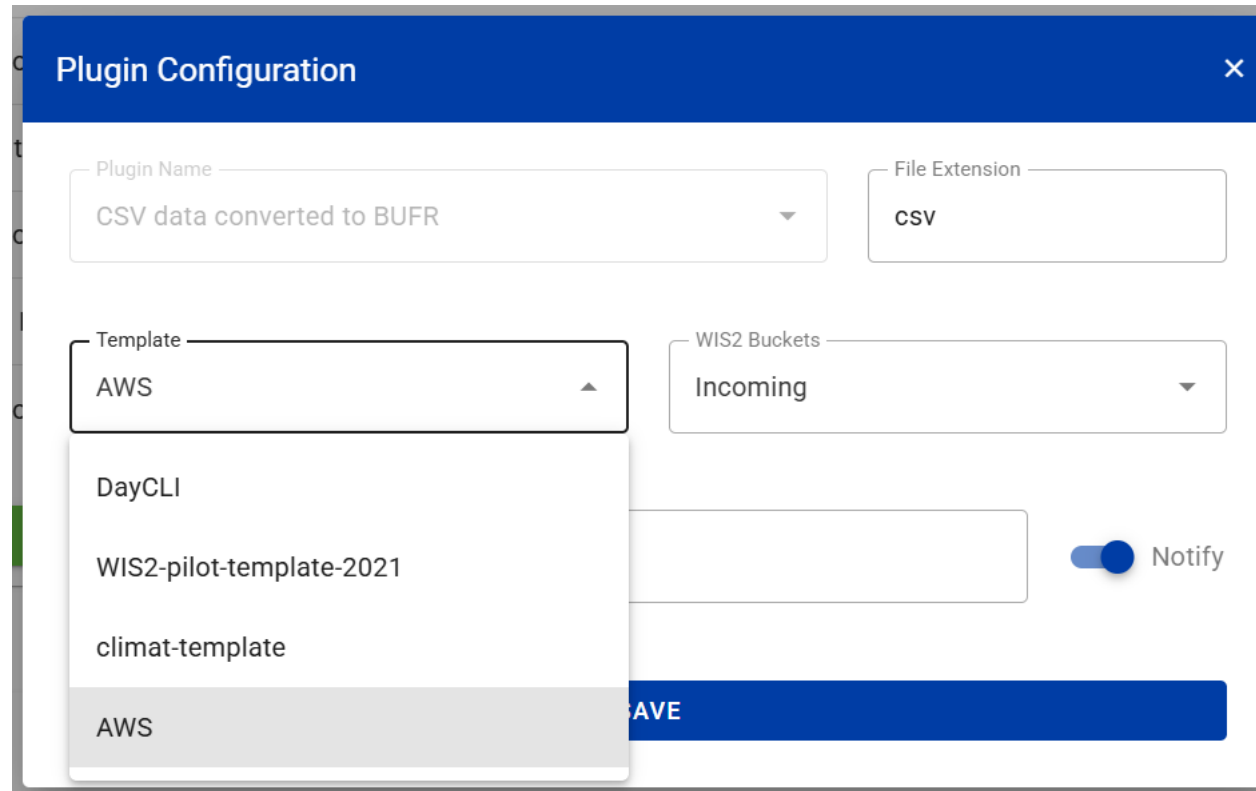
# Creating your own BUFR mapping template for CSV to BUFR

# CSV-to-BUFR mapping templates

The BUFR template mappings included in wis2box are example templates as defined in: [github.com/World-Meteorological-Organization/csv2bufr-templates](https://github.com/World-Meteorological-Organization/csv2bufr-templates)

Built-in templates have **limited use**:

- Input CSV data must be formatted to have the exact column names as defined in the template
- **Only the parameters included in the template can be encoded into BUFR**



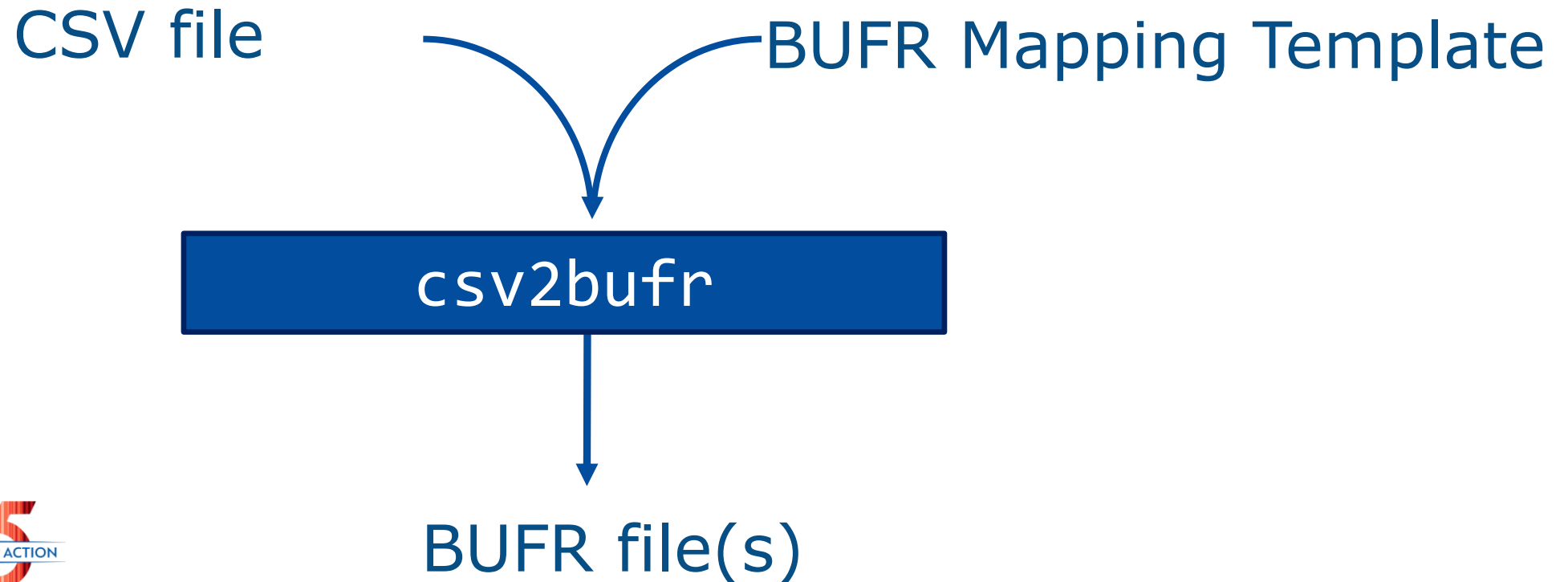
The screenshot shows a 'Plugin Configuration' dialog box with the following fields and options:

- Plugin Name:** A dropdown menu currently showing 'CSV data converted to BUFR'.
- File Extension:** A text input field containing 'CSV'.
- Template:** A dropdown menu currently showing 'AWS'. A list of available templates is open below it, including 'DayCLI', 'WIS2-pilot-template-2021', 'climat-template', and 'AWS' (which is highlighted).
- WIS2 Buckets:** A dropdown menu currently showing 'Incoming'.
- Notify:** A toggle switch that is currently turned on.
- SAVE:** A blue button at the bottom right of the dialog.

# CSV-to-BUFR mapping templates

**Defining your own BUFR Mapping Template enables you to:**

- **Map to column-names specific to your input CSV**
- **Set your own min/max values for quality-control**
- **Encode any parameter defined in [confluence.ecmwf.int/display/ECC/BUFR+tables](https://confluence.ecmwf.int/display/ECC/BUFR+tables)**



# BUFR Tables for ecCodes

[confluence.ecmwf.int/display/ECC/BUFR+tables](https://confluence.ecmwf.int/display/ECC/BUFR+tables)

## 3 types of tables are used:

- **Element tables** translate BUFR descriptors to **human readable ecCodes-key** and define **Units** and **Scale** of the data
- **Sequences tables** define a series of elements
- **Code-flags tables** define controlled vocabularies

Last number of WMO=X table contains all previous table-versions

### ▼ BUFR tables

- BUFR code-flag tables
- BUFR element tables
- BUFR sequence tables

- WMO=36 element table
- WMO=37 element table
- WMO=38 element table
- WMO=39 element table
- WMO=40 element table
- WMO=41 element table
- **WMO=42 element table**

# BUFR Sequence Examples

**BUFR sequence 301011**  
groups 3 elements for Date:

Sequences **301011** - table entries

| Position | FXXYYY |
|----------|--------|
| 1        | 004001 |
| 2        | 004002 |
| 3        | 004003 |

**BUFR sequence 301012**  
groups 2 elements for Time:

Sequences **301012** - table entries

| Position | FXXYYY |
|----------|--------|
| 1        | 004004 |
| 2        | 004005 |

**Information about each element is listed in the BUFR element tables:**

## Class 4 - Location (time)

| FXXYYY | Class (XX) | Code (YYY) | Name   | Key    | Units | Scale | Reference | Width | CREX units | CREX scale | CREX width |
|--------|------------|------------|--------|--------|-------|-------|-----------|-------|------------|------------|------------|
| 004001 | 4          | 1          | YEAR   | year   | a     | 0     | 0         | 12    | a          | 0          | 4          |
| 004002 | 4          | 2          | MONTH  | month  | mon   | 0     | 0         | 4     | mon        | 0          | 2          |
| 004003 | 4          | 3          | DAY    | day    | d     | 0     | 0         | 6     | d          | 0          | 2          |
| 004004 | 4          | 4          | HOUR   | hour   | h     | 0     | 0         | 5     | h          | 0          | 2          |
| 004005 | 4          | 5          | MINUTE | minute | min   | 0     | 0         | 6     | min        | 0          | 2          |
| 004006 | 4          | 6          | SECOND | second | s     | 0     | 0         | 6     | s          | 0          | 2          |

# How to make your own BUFR Mapping Template

Login to wis2box-api docker container:

```
python3 wis2box-ctl.py login wis2box-api
```

Run the “csv2bufr mappings create” command for the BUFR sequences and/or elements you want to map:

```
csv2bufr mappings create 301150 301011 301012 301021 007031 302001 --output  
/data/wis2box/mappings/my_own_template.json
```

Note: the directory /data/wis2box/mappings is mapped to \$WIS2BOX\_HOST\_DATADIR/mappings allowing you to edit the file on the host

# Use custom csv2bufr template in BUFR

You can select the new template in the **Plugin Configuration** of the Dataset Mappings Editor in the wis2box-webapp:

Plugin Configuration

Plugin Name: CSV data converted to BUFR

File Extension: CSV

Template: my\_custom\_template

WIS2 Buckets: Incoming

Notify: ☒

SAVE

ADD A PLUGIN +

Or define the custom BUFR mapping in the csv2bufr plugin configuration in a metadata-yaml MCF as follows:

```
CSV:
- plugin: wis2box.data.csv2bufr.ObservationDataCSV2BUFR
  template: /data/wis2box/mappings/my_custom_template.json # custom BUFR mapping
  notify: true
  file-pattern: '^.*\.csv$'
```

# Debugging custom csv2bufr template in BUFR

Debug your BUFR mapping template using **csv2bufr command line** within the wis2box-api container

Put test\_data.csv in \$WIS2BOX\_HOST\_DATADIR/mappings  
(mapped to /data/wis2box/mappings in wis2box-api container)  
and run the the command `csv2bufr data transform`

```
python3 wis2box-ctl.py login  
csv2bufr data transform --bufr-template /data/wis2box/mappings/my_own_template.json  
/data/wis2box/test_data.csv
```

Check for **output-errors** and edit the custom template as required

Use **bufr\_dump** to inspect the BUFR-file created by the transform and **check all required data is encoded**

# Summary

**Create your own BUFR mapping templates for CSV-to-BUFR conversion enables you to:**

- **Encode any data element defined in the [ecCodes BUFR Tables](#)**
- **Configure your own quality control checks**
- **Define CSV-column names in the mapping template to match your input-data**

The command line tool “`csv2bufr mappings create`” inside the `wis2box-api` container helps you to quickly create a new mappings file, which can be edited to suit your needs

# Thank you

wmo.int



WORLD  
METEOROLOGICAL  
ORGANIZATION

