

# Climate Index Metadata and its Implementation

Klaus Zimmermann   Lars Barring

Rosby Centre  
Sveriges Meteorologiska och Hydrologiska Institut

2020-04-07 EGU Sharing Geosciences Online

# Climate Index Metadata

## Why Metadata?

- ▶ Enable automation and interoperability
- ▶ Improve understandability and user experience

## State of the Art

- ▶ CF Conventions & CMIP Data Request

## Extend for Climate Index Domain

- ▶ Yaml files, generated from
- ▶ Excel spreadsheet

<https://bitbucket.org/cf-index-meta/cf-index-meta>

# An Implementation: Climix

## Another One?

- ▶ Existing: icclim, climact2, xclim, climindex.pcic, ...
- ▶ Metadata & calculation details only in code

## Features

- ▶ Index definition from metadata standard
- ▶ All metadata & calculation details from open standard
- ▶ Extensible via python `entry_points`
- ▶ Iris & Dask: shared & distributed memory parallelization

<https://git.smhi.se/climix/climix>

# Outline

## Climate Index Metadata

Metadata Examples

Contents

## Climix

# CF Index Meta

- ▶ Aspires to be an emerging community standard
- ▶ Current status at  
<https://bitbucket.org/cf-index-meta/cf-index-meta>

# CF Index Meta

## Format

- ▶ Main document: `master_table.xls`
- ▶ Transformed into set of yaml files

## Contents

Three parts of metadata

- ▶ Index definitions
- ▶ Input variable definitions
- ▶ Index functions

# An Example Index

```
txx:
  reference: ETCCDI
  period:
    allowed:
      annual:
      seasonal:
      monthly:
    default: annual
  output:
    var_name: "txx"
    standard_name: air_temperature
    long_name: "Maximum daily maximum temperature"
    units: "degree_Celsius"
    cell_methods:
      - time: maximum within days
      - time: maximum over days
  input:
    data: tasmax
  index_function:
    name: statistics
    parameters:
      reducer:
        kind: reducer
        reducer: max
  ET:
    short_name: "txx"
    long_name: "Maximum daily maximum temperature"
    definition: "Maximum value of daily TX"
```

# An Example Index

txx:

reference: ETCCDI

period:

allowed:

annual:

seasonal:

monthly:

default: annual

output:

var\_name: "txx"

standard\_name: air\_temperature

long\_name: "Maximum daily maximum temperature"

units: "degree\_Celsius"

cell\_methods:

- time: maximum within days

- time: maximum over days

input:

data: tasmax

index\_function:

name: statistics

parameters:

reducer:

kind: reducer

reducer: max

ET:

short\_name: "txx"

long\_name: "Maximum daily maximum temperature"

definition : "Maximum value of daily TX"

index name



# An Example Index

txx:

reference: ETCCDI

period:

allowed:

annual:

seasonal:

monthly:

default: annual

output:

var\_name: "txx"

standard\_name: air\_temperature

long\_name: "Maximum daily maximum temperature"

units: "degree\_Celsius"

cell\_methods:

- time: maximum within days

- time: maximum over days

input:

data: tasmax

index\_function:

name: statistics

parameters:

reducer:

kind: reducer

reducer: max

reference information

ET:

short\_name: "txx"

long\_name: "Maximum daily maximum temperature"

definition : "Maximum value of daily TX"

# An Example Index

txx:

reference: ETCCDI

period:

allowed:

annual:

seasonal:

monthly:

default: annual

output:

var\_name: "txx"

standard\_name: air\_temperature

long\_name: "Maximum daily maximum temperature"

units: "degree\_Celsius"

cell\_methods:

- time: maximum within days

- time: maximum over days

input:

data: tasmax

index\_function:

name: statistics

parameters:

reducer:

kind: reducer

reducer: max

ET:

short\_name: "txx"

long\_name: "Maximum daily maximum temperature"

definition : "Maximum value of daily TX"

allowable and standard period

# An Example Index

```
txx:
  reference: ETCCDI
  period:
    allowed:
      annual:
      seasonal:
      monthly:
    default: annual
  output:
    var_name: "txx"
    standard_name: air_temperature
    long_name: "Maximum daily maximum temperature"
    units: "degree_Celsius"
    cell_methods:
      - time: maximum within days
      - time: maximum over days
  input:
    data: tasmax
  index_function:
    name: statistics
    parameters:
      reducer:
        kind: reducer
        reducer: max
  ET:
    short_name: "txx"
    long_name: "Maximum daily maximum temperature"
    definition : "Maximum value of daily TX"
```

to be used in output file

# An Example Index

```
txx:
  reference: ETCCDI
  period:
    allowed:
      annual:
      seasonal:
      monthly:
    default: annual
  output:
    var_name: "txx"
    standard_name: air_temperature
    long_name: "Maximum daily maximum temperature"
    units: "degree_Celsius"
    cell_methods:
      - time: maximum within days
      - time: maximum over days
  input:
  data: tasmx
  index_function:
    name: statistics
    parameters:
      reducer:
        kind: reducer
        reducer: max
  ET:
    short_name: "txx"
    long_name: "Maximum daily maximum temperature"
    definition : "Maximum value of daily TX"
```

input data to be operated on  
refers to variable metadata  
(see below)

# An Example Index

```
txx:
  reference: ETCCDI
  period:
    allowed:
      annual:
      seasonal:
      monthly:
    default: annual
  output:
    var_name: "txx"
    standard_name: air_temperature
    long_name: "Maximum daily maximum temperature"
    units: "degree_Celsius"
    cell_methods:
      - time: maximum within days
      - time: maximum over days
  input:
    data: tasmax
  index_function:
    name: statistics
    parameters:
      reducer:
        kind: reducer
        reducer: max
  ET:
    short_name: "txx"
    long_name: "Maximum daily maximum temperature"
    definition : "Maximum value of daily TX"
```

determines calculation

# Input Data

tasmax:

standard\_name: air\_temperature

cell\_methods:

- time: maximum

aliases:

- tasmaxadjust
- tmax
- tx
- maxt
- TMAX
- Tmax
- TX
- MAXT
- maxT

# Index Function

spell\_length :

description : |

Calculates statistics on lengths of spells .

First , the threshold is transformed to the same standard\_name and units as the input data.

Then the thresholding is performed as condition(data, threshold), ie if condition is <, data < threshold.

Then the spells are determined, and finally the statistics according to the specified reducer are calculated.

parameters:

threshold:

kind: quantity

condition:

kind: operator

reducer:

kind: reducer

# Contents

Reference	Number of indices	
	total	ready
ETCCDI	27	18
ET-SCI	33	13
ECA&D	44	23
CLIPC	52	40
B4EST	2	0
SMHI	1	1
-	5	5
total	164	100

- ▶ **ready**: all metadata in place
- ▶ Number of index functions: 12



# Outline

## Climate Index Metadata

Metadata Examples

Contents

## Climix

# Overview

- ▶ Python package to calculate climate indices
- ▶ Index definition directly from metadata
- ▶ Based on Iris and Dask
- ▶ Version 1.0.0 expected by July 2020
- ▶ Technology preview at  
<https://git.smhi.se/climix/climix>

# Index Definitions

- ▶ Directly read from metadata yaml files
- ▶ Add new, custom indices by adding yaml files; no coding needed

# Architecture I

## Climix...

- ▶ Implements index functions
- ▶ Sets up Dask environment
- ▶ Reads input data
- ▶ Stores output

## Architecture II

### entry\_points for Index Functions

- ▶ Connect metadata and implementation
- ▶ Easily add new or alternative implementations

```
setuptools.setup(  
    entry_points={  
        'climix.index_functions': [  
            'spell_length=climix.index_functions:SpellLength',  
        ],  
    },  
)
```

# Summary

## CF Index Meta

- ▶ Emerging community standard
- ▶ Build on CF and extend it

## Climix

- ▶ Open Source, Python climate index package
- ▶ Strongly metadata oriented

