

**Distributed Oceanographic Match-up Service (DOMS)
Translation Specification: SPURS In Situ Data**

Nga Quach and Thomas Huang

Jet Propulsion Laboratory/California Institute of Technology, Pasadena, CA

Contact: Thomas.Huang@jpl.nasa.gov

Introduction

To make the DOMS matchup output meaningful and easy to use the collaborating partners must standardize their data within the DOMS prototype. This document defines the translation of SPURS data from their native format at JPL into Apache Solr (the in-situ indexing approach chosen for DOMS).

Date and time:

Convert SPURS NetCDF time convention value to ISO 8601.

SPURS NetCDF form: seconds since 1970-01-01 00:00:00 0:00

ISO 8601 form: YYYY-MM-DDThh:mm:ssZ

CF name = time

Latitude:

Copy lat in decimal degrees with +North and -South

CF name = latitude

Longitude:

Decimal degrees in -180 West to +180 East

Convert lat 0.00 to 359.99 to -179.99 to 180.00 East

Note there will be no 180.00 West.

CF name = longitude

Platform:

DOMS Index Code	DOMS Description	SPURS PT mapping
1	ship	SPURS1_CTD, SPURS1_UCTD, SPURS1_TSG, SPURS1_METEO, SPURS1_ADCP
2	moored surface buoy	SPURS1_MOORING_WHOI, SPURS1_MOORING_PICO
3	drifting surface float	SPURS1_DRIFTER

4	drifting subsurface profiling float	SPURS1_ARGO, SPURS1_FLOAT_NEUTRALLYBUOYANT
5	autonomous underwater vehicle	SPURS1_ECOMAPPER, SPURS1_TENUSEGLIDER, SPURS1_WAVEGLIDER, SPURS1_SEAGLIDER
8	towed unmanned submersible	SPURS1_SEASOAR

Device:

DOMS Index Code	DOMS Description	SPURS Device (from PT) mapping
3	CTD	SPURS1_CTD, SPURS1_UCTD, SPURS1_ARGO, SPURS1_SEAGLIDER, SPURS1_MOORING_WHOI, SPURS1_MOORING_PICO, SPURS1_SEASOAR, SPURS1_ECOMAPPER, SPURS1_TENUSEGLIDER, SPURS1_SEAGLIDER, SPURS1_WAVEGLIDER, SPURS1_DRIFTER (surface T/S only), SPURS1_FLOAT_NEUTRALLYBUOYANT
4	Current profilers / acousticDopplerCurrentProfiler	SPURS1_ADCP

Mission:

Not recorded

Sea_water_temperature:

Set as value from the following variable, whichever exists, from the following list:
temperature, temperature1, TEMP, temperature_ctd

Units: degrees_C

CF name = sea_water_temperature

Sea_water_temperature_depth:

If sea_water_temperature is available, set as value from the following variable, whichever exists, from the following list: z_ctd, DEPTH, depth

If depth measurement is not provided in data file, it is calculated from pressure variable, PRES, and latitude using

<http://pythonhosted.org/seawater/eos80.html#seawater.eos80.dpth>

CF name = sea_water_temperature_depth

Sea_water_temperature_quality:

Not recorded in data file. All values in data file are good.

CF name = sea_water_temperature_quality

Sea_water_salinity:

Set as value from the following variable, whichever exists, from the following list:

salinity, salinity1, salinity_ctd

Units: psu

CF name = sea_water_salinity

Sea_water_salinity_depth:

If sea_water_salinity is available, set as value from the following variable, whichever exists, from the following list: z_ctd, DEPTH, depth

If depth measurement is not provided in data file, it is calculated from pressure variable, PRES, and latitude using

<http://pythonhosted.org/seawater/eos80.html#seawater.eos80.dpth>

CF name = sea_water_salinity_depth

Sea_water_salinty_quality:

Not recorded in data file. All values in data file are good.

CF name = sea_water_salinty_quality

Wind_speed:

Set as value from RELWSPD variable if it exists

Units: m/s

CF Name = wind_speed

Eastward_wind and Northward_wind:

Use RELWSPD and RELWDIR to derive

Convert degrees to radians

```
wind_dir_rad = numpy.radians(dataset[wind_dir_var])
```

U component

```
dataset['eastward_wind'][:] = numpy.multiply(dataset[wind_speed_var],
numpy.sin(wind_dir_rad))
```

V component

```
dataset['northward_wind'][:] = numpy.multiply(dataset[wind_speed_var],
numpy.cos(wind_dir_rad))
```

CF names = eastward_wind and northward_wind

Wind_depth:

If wind_speed is available, set as value from the following variable, whichever exists, from the following list: z_ctd, DEPTH, depth

If depth measurement is not provided in data file, it is calculated from pressure variable, PRES, and latitude using

<http://pythonhosted.org/seawater/eos80.html#seawater.eos80.dpth>

CF Name = wind_depth

Wind_quality:

Not recorded in data file. All values in data file are good.

CF name = wind_quality

Meta:

Not recorded

CF name = meta