Ad-hoc Processing with the CoastWatch Utilities

Peter Hollemans, Terrenus Earth Sciences Consultant for NOAA/NESDIS

CoastWatch Operations Managers Meeting, June, 2007

Talk Outline

- Extracting Information
- Data Processing
- Custom Code

File Contents

Extracting Information

Data Processing

Custom Code

- Dump file contents with the cwinfo tool
- Extract raw metadata values from HDF files with the hdatt tool

```
phollema@bean<Data> cwinfo 2006_249_2137_n18_wn.hdf
Contents of file 2006_249_2137_n18_wn.hdf

Global information:
  Satellite:            noaa-18
  Sensor:               avhrr
  Date:                 2006/09/06 JD 249
  Time:                 21:37:04 UTC
  Scene time:           day
  Projection type:      mapped
  Map projection:       Mercator
  Map affine:           0 -1470 1470 0 -15028629.69 6343200.92
  Spheroid:             WGS 84
  Origin:               USDOC/NOAA/NESDIS CoastWatch
  Format:               CoastWatch HDF version 3.4

Variable information:
  Variable        Type    Dimensions  Units           Scale     Offset
  avhrr_ch1       short   1024x1024   percent         0.01      0
  avhrr_ch2       short   1024x1024   percent         0.01      0
  avhrr_ch3       short   1024x1024   celsius         0.01      0
  avhrr_ch4       short   1024x1024   celsius         0.01      0
  avhrr_ch5       short   1024x1024   celsius         0.01      0
  cloud           ubyte   1024x1024   -               1         0
  graphics        ubyte   1024x1024   -               1         0
  rel_azimuth     short   1024x1024   degrees         0.01      0
  sat_zenith      short   1024x1024   degrees         0.01      0
  sst             short   1024x1024   celsius         0.01      0
  sun_zenith      short   1024x1024   degrees         0.01      0
```

```
phollema@bean<Data> hdatt -h
Usage: hdatt [OPTIONS] input
       hdatt {-n, --name=STRING} [OPTIONS] input
       hdatt {-n, --name=STRING} {-l, --value=STRING1[/STRING2/...]}
         [OPTIONS] input
Reads or writes HDF file attributes.

Main parameters:
  -l, --value=STRING          Set value to write to attribute.
  -n, --name=STRING           Set name of attribute to read or write.
  input                       The input data file name.

Options:
  -h, --help                  Show this help message.
  -t, --type=TYPE             Set attribute value type for writing.  TYPE
                               may be 'string', 'byte', 'short', 'int',
                               'long', 'float', or 'double'.
  -V, --variable=STRING       Set variable to which attribute belongs.
  --version                   Show version information.
```

```
phollema@bean<Data> hdatt 2006_249_2137_n18_wn.hdf
satellite = noaa-18
sensor = avhrr
origin = USDOC/NOAA/NESDIS CoastWatch
cwhdf_version = 3.4
pass_type = day
pass_date = 13397
start_time = 77824.0
projection_type = mapped
projection = Mercator
gctp_sys = 5
gctp_zone = 0
gctp_parm = 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
gctp_datum = 12
et_affine = 0.0 -1470.0 1470.0 0.0 -1.5028629694237337E7 6343200.918551117
rows = 1024
cols = 1024
polygon_latitude = 49.59106659023744 49.59106659023744 49.59106659023744
49.59106659023744 49.59106659023744 47.34572341754 52 44.99999999987 04
42.55345611162261 40.00635240249667 40.00635240249667 40.00635240249667
40.00635240249667 40.00635240249667 42.55345611162261 44.99999999987 04
47.34572341754 52 49.59106659023744
polygon_longitude = -135.00447753705 89 -131.62723876852945 -128.25
-124.87276123147056 -121.49552246294114 -121.49552246294114 -121.49552246294114
-121.49552246294114 -121.49552246294114 -124.87276123147056 -128.25
-131.62723876852945 -135.00447753705 89 -135.00447753705 89 -135.00447753705 89
-135.00447753705 89 -135.00447753705 89
history = [cwf 3.1.10-pre] cwimport product.tshdf product.hdf
```

```
phollema@bean<Data> hdatt --name rows 2006_249_2137_n18_wn.hdf
1024

phollema@bean<Data> hdatt --name satellite 2006_249_2137_n18_wn.hdf
noaa-18

phollema@bean<Data> hdatt --variable avhrr_ch2 2006_249_2137_n18_wn.hdf
long_name = avhrr_ch2
units = percent
coordsys = Mercator
_FillValue = -32768
missing_value = -32768
scale_factor = 0.01
scale_factor_err = 0.0
add_offset = 0.0
add_offset_err = 0.0
calibrated_nt = 0
fraction_digits = 2

phollema@bean<Data> hdatt -V avhrr_ch2 -n scale_factor 2006_249_2137_n18_wn.hdf
0.01
```

Data Statistics

Extracting Information

Data Processing

Custom Code

- Compute statistics with the cwstats tool
- Select only certain variables
- Sample only some of the data

```
phollema@bean<Data> cwstats -h
Usage: cwstats [OPTIONS] input
Calculates a number of statistics for each variable in an Earth data file.

Main parameters:
  input                       The input data file name.

Options:
  -h, --help                  Show this help message.
  -i, --region=LAT/LON/RADIUS
                              Only compute statistics for data values
                               within so many kilometers of a location.
  -l, --limit=STARTROW/STARTCOL/ENDROW/ENDCOL
                              Only compute statistics for data values
                               between the limits.
  -m, --match=PATTERN         Only compute statistics for variables
                               matching the pattern.
  -s, --stride=N              Sample every Nth value in each dimension.
  -S, --sample=FACTOR         Sample only a fraction of the data in each
                               variable.  FACTOR must be between 0 and 1.
  --version                   Show version information.
```

```
phollema@bean<Data> cwstats --match 'avhrr_ch[124]' --sample 0.01
2006_249_2137_n18_wn.hdf
Variable        Count     Valid     Min         Max         Mean        Stdev
avhrr_ch1       10609     10609     1.79        51          5.834375    7.079225
avhrr_ch2       10609     10609     0.64        53.6        6.710064    7.978374
avhrr_ch4       10609     10609     -34.82      51.53       16.240626   6.19136
```

Data Sampling

Extracting Information

Data Processing

Custom Code

- Sample data with the cwsample tool
- Select only certain variables to sample
- Put a header on the output columns

```
phollema@bean<Data> cwsample -h
Usage: cwsample {-s, --sample=LATITUDE/LONGITUDE} [OPTIONS] input output
       cwsample {-S, --samples=FILE} [OPTIONS] input output
Extracts data values at specified Earth locations from 2D data variables.

Main parameters:
  -s, --sample=LATITUDE/LONGITUDE
                              Sample at single Earth location.
  -S, --samples=FILE          Sample at multiple Earth locations.
  input                       The input data file name.
  output                      The output text file name or '-' for stdout.

Options:
  -d, --dec=DECIMALS          Set decimal places in geographic coordinates.
  -D, --delimit=STRING        Set delimiter between data columns.
  -h, --help                  Show this help message.
  -H, --header                Print one line header on output table.
  -i, --imagecoords           Print image row and column coordinates.
  -m, --match=PATTERN         Sample only variables matching the pattern.
  -M, --missing=VALUE         Set value to print for missing data.
  -n, --nocoords              Do not print geographic coordinates.
  -R, --reverse               Reverse order of geographic coordinates.
  -V, --variable=NAME1[/NAME2/...]
                              Sample only variable names listed in order.
  --version                   Show version information.
```

```
phollema@bean<Data> cat samples.txt
48 -126
48 -126.5
48 -127
48 -127.5
48 -128

phollema@bean<Data> cwsample --missing -999 --header --variable sst/cloud
--samples samples.txt 2006_249_2137_n18_wn.hdf output.txt

phollema@bean<Data> cat output.txt
latitude longitude sst cloud
48 -126 14.3 -999
48 -126.5 13.77 -999
48 -127 15.96 -999
48 -127.5 15.12 -999
48 -128 16.6 -999
```

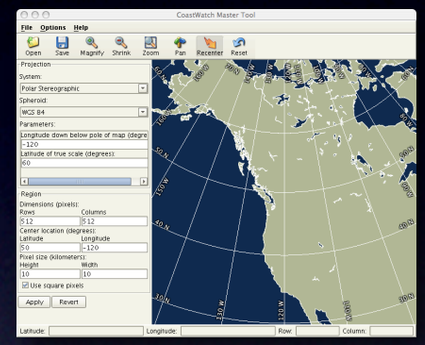
Reprojection

Extracting Information

Data Processing

Custom Code

- Define projection with the CoastWatch Master Tool



- Register data to the projection with cwregister tool
- Reprojection works on swath data (like AVHRR) and mapped data (like Mercator)

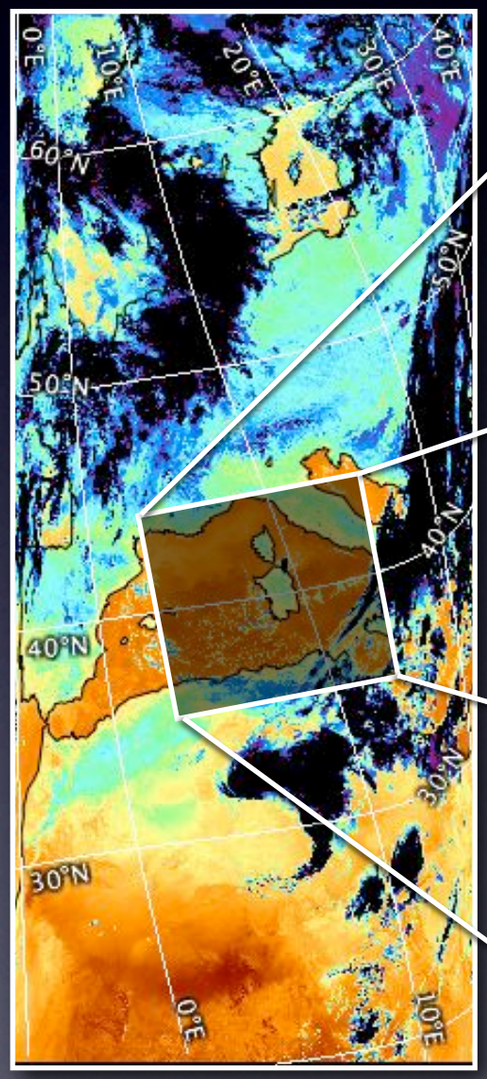
Reprojection Example

Extracting Information

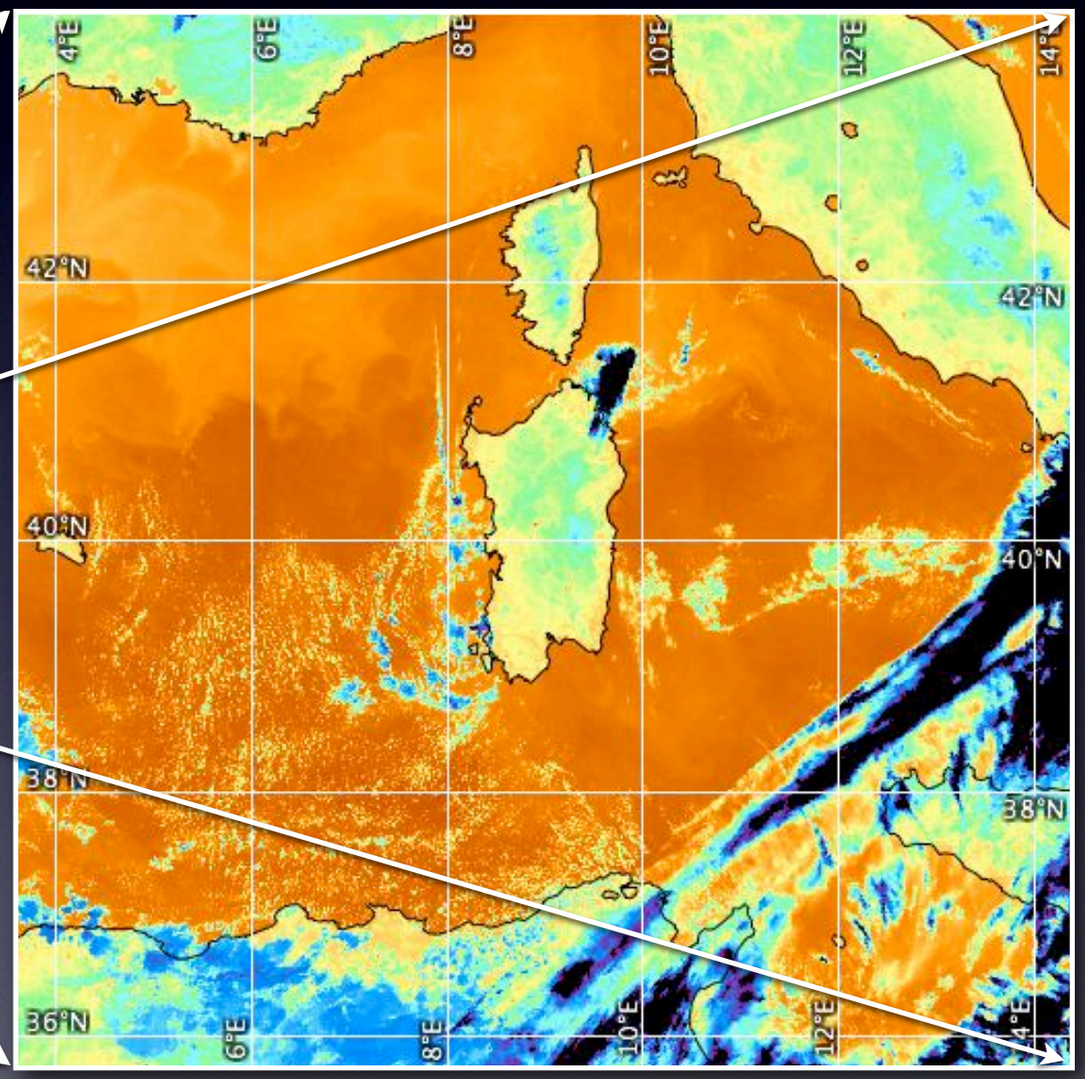
Data Processing

Custom Code

Swath



Mercator



```
phollema@bean<Data> cwregister -v --match 'avhrr_ch[45]' med.hdf 06100601.hdf
06100601_med.hdf
cwregister: Reading master med.hdf
cwregister: Reading input 06100601.hdf
cwregister: Creating output 06100601_med.hdf
cwregister: Adding avhrr_ch4 to resampled grids
cwregister: Adding avhrr_ch5 to resampled grids
class noaa.coastwatch.util.InverseGridResampler: Found 2 grid(s) for resampling
class noaa.coastwatch.util.InverseGridResampler: Resampling to 1200x1200 from
4672x2048
class noaa.coastwatch.util.InverseGridResampler: Creating location estimators
class noaa.coastwatch.util.InverseGridResampler: Working on output row 0
class noaa.coastwatch.util.InverseGridResampler: Working on output row 100
class noaa.coastwatch.util.InverseGridResampler: Working on output row 200
class noaa.coastwatch.util.InverseGridResampler: Working on output row 300
class noaa.coastwatch.util.InverseGridResampler: Working on output row 400
class noaa.coastwatch.util.InverseGridResampler: Working on output row 500
class noaa.coastwatch.util.InverseGridResampler: Working on output row 600
class noaa.coastwatch.util.InverseGridResampler: Working on output row 700
class noaa.coastwatch.util.InverseGridResampler: Working on output row 800
class noaa.coastwatch.util.InverseGridResampler: Working on output row 900
class noaa.coastwatch.util.InverseGridResampler: Working on output row 1000
class noaa.coastwatch.util.InverseGridResampler: Working on output row 1100
cwregister: Closing files
```

Computation

Extracting Information

Data Processing

Custom Code

- Compute math expressions with with the cwmath tool:
  - data masking
  - solar zenith angle correction
  - NDVI, SST, etc
- Combine data files with the cwcomposite tool

```
phollema@bean<Data> cwmath -v --expr "mcsst = 0.98*(avhrr_ch4+273.15) + 2.77*
(avhrr_ch4-avhrr_ch5) + 0.44*(avhrr_ch4-avhrr_ch5)*(1/cos(sat_zenith*pi/180) -
1) - 266.29" --template avhrr_ch4 2007_161_1050_n15_sr.hdf
2007_161_1050_n15_sr_mcsst.hdf
cwmath: Opening input 2007_161_1050_n15_sr.hdf
cwmath: Creating output 2007_161_1050_n15_sr_mcsst.hdf
cwmath: Creating mcsst variable
cwmath: Computing row 0
cwmath: Computing row 100
cwmath: Computing row 200
cwmath: Computing row 300
cwmath: Computing row 400
cwmath: Computing row 500
cwmath: Computing row 600
cwmath: Computing row 700
cwmath: Computing row 800
cwmath: Computing row 900
cwmath: Computing row 1000
cwmath: Computing row 1100
cwmath: Computing row 1200
```

Data Interfaces

Extracting Information

Data Processing

Custom Code

- Call command line tools from a scripting language: Bash, Perl, Python
- Use the Java API for direct access to data

```
phollema@bean<Data> cat process.sh
#!/bin/sh

time=`cwinfo $1 | awk '/Scene time:/ { print $3 }'`
if [ "$time" = "day" ] ; then
  echo "Processing daytime data ..."
  # Insert some operation here
elif [ "$time" = "night" ] ; then
  echo "Processing nighttime data ..."
  # Insert some operation here
else
  echo "The scene time is unknown!"
  exit 2
fi

phollema@bean<Data> ./process.sh 2006_249_2137_n18_wn.hdf
Processing daytime data ...
```

```
phollema@bean<Data> cat FormatDetect.java
import java.io.*;
import noaa.coastwatch.io.*;

public class FormatDetect {
  public static final void main (String[] argv) throws IOException {

    EarthDataReader reader = EarthDataReaderFactory.create (argv[0]);
    System.out.println ("Data file has format " + reader.getDataFormat());

  }
}

phollema@bean<Data> javac FormatDetect.java

phollema@bean<Data> java FormatDetect 2006_352_2044_n18_mo.lac
Data file has format NOAA 1b version 5
```

File Interfaces

Extracting Information

Data Processing

Custom Code

- C language API for HDF
- IDL & Matlab APIs for HDF
- Java API for netCDF (OPeNDAP datasets as well)

```
phollema@bean<Data> cat get_hdf_vars.c
#include <stdio.h>
#include "mfhdf.h"

int main (int argc, char *argv[]) {

  int32 sdId, sdsId;
  int32 nVars, nAtts, index, rank, dimSizes[MAX_VAR_DIMS], dataType;
  intn status;
  char name[MAX_NC_NAME];

  sdId = SDstart (argv[1], DFACC_READ);
  status = SDfileinfo (sdId, &nVars, &nAtts);
  for (index = 0; index < nVars; index++) {
    sdsId = SDselect (sdId, index);
    status = SDgetinfo (sdsId, name, &rank, dimSizes, &dataType, &nAtts);
    printf ("%s\n", name);
    status = SDendaccess (sdsId);
  }

  status = SDend (sdId);

}
```

```
phollema@bean<Data> gcc -o get_hdf_vars get_hdf_vars.c -Ihdf/include -Lhdf/lib
-lmfhdf -ldf -lz -L/sw/lib -ljpeg.62

phollema@bean<Data> ./get_hdf_vars 2006_249_2137_n18_wn.hdf
avhrr_ch1
avhrr_ch2
avhrr_ch3
avhrr_ch4
avhrr_ch5
cloud
graphics
rel_azimuth
sat_zenith
sst
sun_zenith
```

Summary

- Extracting Information: file contents, data statistics and sampling
- Data Processing: reprojection, computation
- Custom Code: data and file interfaces

Acknowledgements

- CoastWatch node and central operations staff
- CoastWatch data users