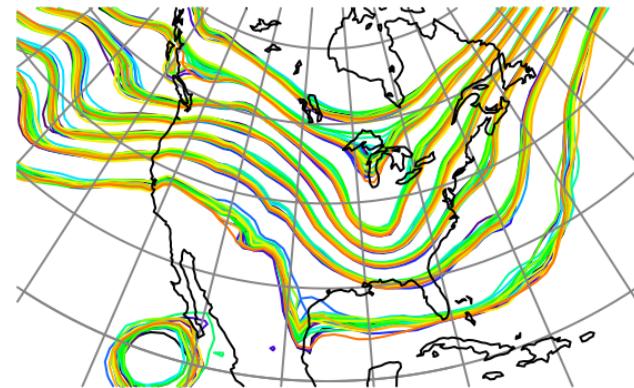


D
A
R
T

ata
ssimilation
esearch
estbed



DART Tutorial Section 17: Creating Observation Sequences



©UCAR 2014

The National Center for Atmospheric Research is sponsored by the National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

NCAR | National Center for
UCAR Atmospheric Research



Types of Observations used in DART

A large number of geophysical observations are already being assimilated into models using DART.

They include:

Atmospheric Observations,

Ocean Observations,

Solar, Space Weather, Extraterrestrial Observations,

Land Observations

Sea ice observations

DART Atmospheric Observations (1)

U,V,T,Q	NCEP: Radiosonde, AIRCRAFT (commercial), ACARS
U,V	NCEP: Cloud Drift Winds from satellite
U,V (ocean surface)	QUIKSCAT, including L2B (JPL)
T,Q,refractivity of the atmosphere	COSMIC Global Positioning Satellite radio occultation
T,Q,Tsurface	AIRS from Aqua/A-train satellite
U,V,T,Q,Tsurface, pressure,altimeter	MADIS: ACARS, Marine and MESONET surface, METAR, radiosonde, satellite wind
Radar reflectivity, radial velocity	NCEP

DART Atmospheric Observations (2)

U,V	MADIS; Wind Profilers, Atmospheric Motion Vectors (AMVs)
U,V,T,Q,altimeter	OK mesonet (U. OK)
Cloud Liquid Water Path, Cloud Top and Base Pressures	GOES satellite, CIMSS
U,V	SSEC (U Wisconsin): Cloud Drift Winds from satellite
CO (carbon monoxide)	MOPITT
U,V	GOES CIMSS (U. WI); rapid-scan AMVs (Atmospheric Motion Vectors), satellite cloud winds

DART Atmospheric Observations (3)

T,Q,Total Precipitable Water	GOES CIMSS hyperspectral AIRS IR
Total Precipitable Water	AMSR, MODIS Microwave
U,V	Operational typhoon bogus winds, Taiwan Central Weather Bureau
U,V (at wind turbine hub height)	Seimens(?)
Electron density	COSMIC/FORMOSAT-3
U,V,T	GTS
Chemical concentrations	IASI on EUMETSAT Polar System MetOp satellite
Aerosol optical depth (AOD)	TERA and AQUA

DART Solar, Space Weather, Extraterrestrial Obs

Radiances, Occultation on Mars	TES, limb sounder on Mars
Density, ion concentrations	CHAMP
Thermospheric Mass Densities	CHAMP, GRACE
Electron densities	COSMIC
Total Electron Density	Garner GPS Archive
Orbital element information	NORAD
Solar Magnetic Fields	Wilcox, Mt Wilson, National Solar Observatories
Rotational, Meridional Circulation	Mt Wilson, SoHO, SDO, HMI

DART Ocean Observations

T, Salinity	World Ocean Database: Argo floats, CTD(ships), XBT, moored thermistors, drifting buoys(GT-SPP)
Surface U, V currents	CODAR

DART Land Observations

Snow cover	MODIS
Leaf area index	MODIS
Total water storage	GRACE
Brightness temperature	AMSR-E
Heat Flux, Net Carbon	Ameriflux tower network
Soil Moisture	COSMOS (neutron counter)

Building Real Observation Sequences

In the DART distribution, *observations/obs_converters* contains a collection of conversion programs from a variety of formats to DART obs_seq format.

See the observation converter overview in:

[observations/obs_converters/observations.html](#)

To create new converters for your data:

File format	<i>A good file to use as a starting point</i>
netCDF	Start with <i>MADIS/convert_madis_profiler.f90</i>
Comma separated text	Start with <i>Ameriflux</i>
Generic text	Start with <i>text</i>
HDF-EOS	Start with <i>AIRS</i>
BUFR or prepBUFR	Start with <i>NCEP</i>
Dense data, e.g. Satellite Swaths	Start with <i>tpw</i>
Ray-path integrated data	Start with <i>gps</i>
World Ocean Database packed ASCII	Start with <i>WOD</i>

Observation Sequence Tools

These programs are found in directory *assimilation_code/programs*.

Each program has its own directory that includes source code (*f90*), a default namelist (*nmf*, if required), and documentation (*html*).

<code>obs_diag</code>	Primary tool for evaluating obs space performance.
<code>obs_seq_to_netcdf</code>	Converts an <code>obs_seq</code> file to netcdf format. Cannot propagate per-obs-type metadata.
<code>obs_sequence_tool</code>	General tool with powerful capabilities
<code>obs_common_subset</code>	Select same obs from multiple files
<code>obs_seq_coverage</code>	Identify the same obs locations through a time series
<code>obs_seq_verify</code>	Creates a netCDF file for use in verification studies
<code>obs_selection</code>	Select a subset of observations based on another <code>obs_seq</code> file or the output of the coverage tool
<code>obs_loop</code>	Template that is a good starting point to write your own <code>obs_sequence</code> modifying programs.

Structure of an Obs Sequence File

See the online documentation:

http://www.image.ucar.edu/DARes/DART/DART2_Observations.php#obs_seq_overview

Can contain multiple data values per observation, and multiple quality control values.

Can contain additional metadata per observation type; e.g. Radar observations, GPS obs. The forward operator code can use it when computing expected values. Must write your own read/write routines which are called by DART code to handle the metadata.

Entries DO NOT have to be physically listed in time order; DART read routines return the next observation, by time, automatically. Saves reordering the entries in the file when inserting new observations.

Structure of an Obs Sequence File (cont)

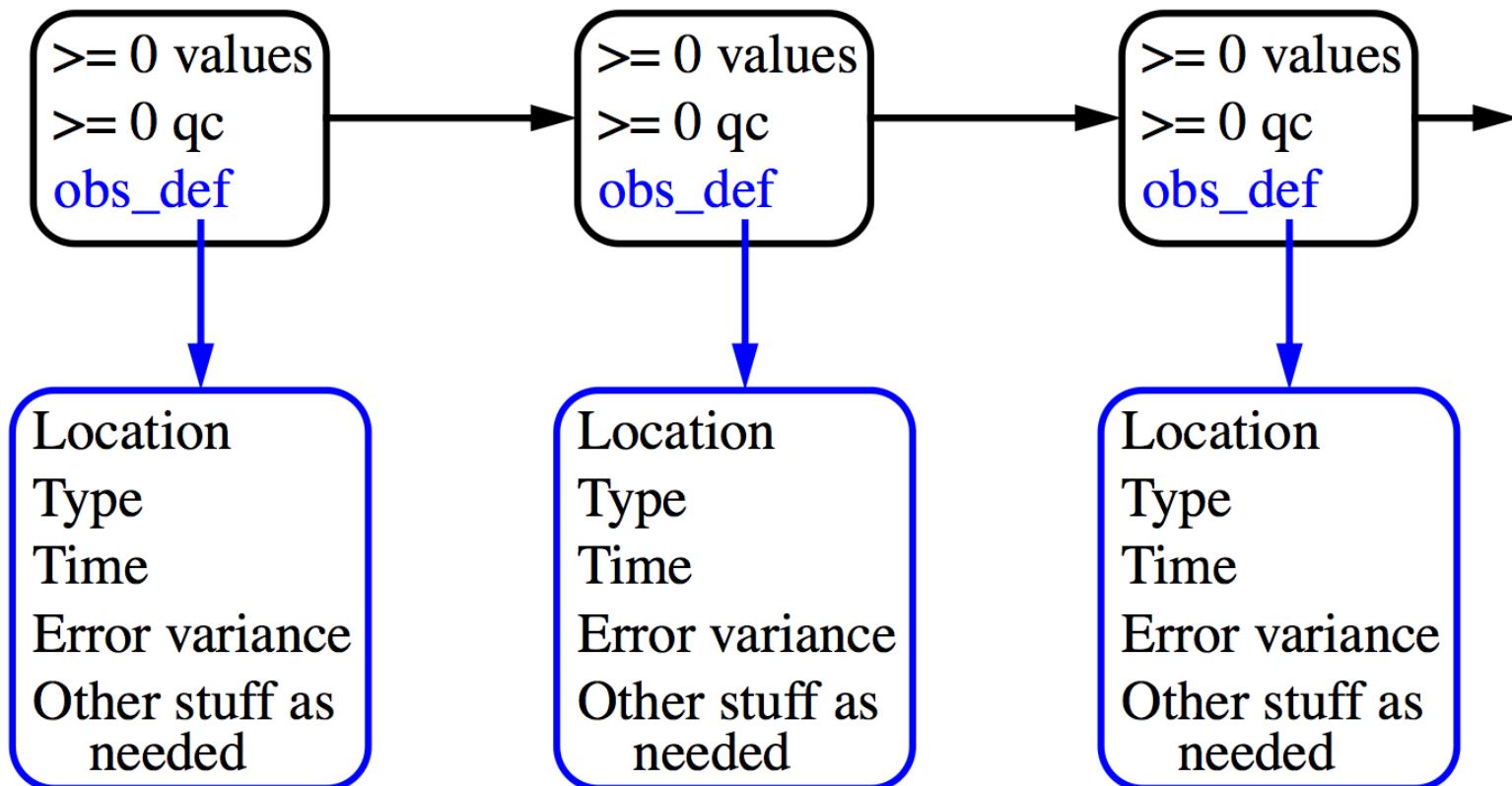
Contains a ‘table of contents’ of observation types at the start:

```
obs_sequence
obs_kind_definitions          (note: these are really a list of obs types)
    8
    25 LAND_SFC_U_WIND_COMPONENT
    26 LAND_SFC_V_WIND_COMPONENT
    27 LAND_SFC_TEMPERATURE
    35 DOPPLER_RADIAL_VELOCITY
    36 RADAR_REFLECTIVITY
    37 RADAR_CLEARAIR_REFLECTIVITY
    61 LAND_SFC_DEWPOINT
    73 LAND_SFC_ALTIMETER
```

The observations are identified by number in the rest of this file, but the numbers DO NOT have to be the same from file to file. The processing is done by matching the string name of the observation.

Structure of an Obs Sequence File (cont)

Sequence contains non-decreasing times in definitions.



Building Real Observation Sequences

Building Real observation sequences:

1. Interactive direct construction: program *create_obs_sequence*
 - Queries for information for each observation in turn.
 - Enter type, location, time, error variance, value, qc value(s).
 - Often convenient to create an input file via an editor or a script.
 - Then redirect this file to standard input for *create_obs_sequence*.
2. Use an existing converter
3. Create your own program
 - The *obs_sequence* module provides full set of Fortran 90 routines to read, write, query, create new, and alter existing observations.
 - (see directory *assimilation_code/modules/observations*)

Creating Synthetic Observation Sequences

Synthetic Observation Sequences are often used for Observing System Simulation Experiment (OSSEs)

Step 1: Create an observation sequence with no values.

A. Direct use of *create_obs_sequence*: no need to specify value for obs.

OR...

B. Synthetic observing network fixed in time:

1. First, use *create_obs_sequence* to specify observations in fixed network, all with time 0 days, 0 seconds.
2. Use *create_fixed_network_seq* to specify times at which fixed network is observed.
3. Times can be regularly or irregularly spaced.

Creating Synthetic Observation Sequences

Creating Synthetic Observation Sequences (cont)

Step 2: Use program *perfect_model_obs* to add observed values.

1. Integrates model.
2. Applies forward operators to get ‘true’ observed values.
3. Adds sample from observational error to get observed value.
(Output *obs_sequence* has **2 values** for each observation, with and without the added error sample).

Step 3: Run the *filter*.

Ensemble mean, spread, (and individual ensemble members if requested using *num_output_obs_members* in *filter_nml*) are added as values in the *obs_sequence*.

This 3 step process was done for all low-order examples so far in the tutorial.

Creating Synthetic Observation Sequences

Program	Usual Filename	Obs Values
	Standard input	not an obs_sequence
create_obs_sequence	set_def.out	0
create_fixed_network_sequence	obs_seq.in	0
perfect_model_obs	obs_seq.out	true, observed
filter	obs_seq.final	true, observed, ensemble mean, spread, ensemble member forward operators

Example: Localized Obs Set for Lorenz 96

1. Run *create_obs_sequence* in the *work* directory.
 - Select 5 observations as upper bound
 - Select 0 copies of data (we'll let *perfect_model_obs* fill these in)
 - Also 0 quality control fields
 - Never input a -1 to terminate (we'll do all 5)
 - Select raw state variable for each observation
 - Pick a location (try grouping them close to 0.5)
 - Select time as 0 days, 0 seconds
 - Error variance of 1.0 for first try
 - Repeat for all 5 observations just varying location
 - Enter *set_def.out* for file name

Example: Localized Obs Set, L96 (cont)

2. Run *create_fixed_network_seq* to observe these 5 obs. repeatedly
 - File is *set_def.out*
 - Select a regularly repeating sequence
 - Select 1000 times
 - Initial time as 0 days, 0 seconds
 - Observation period as 1 hour (0 days, 3600 seconds)
 - Resulting *obs_seq.in* observes once an hour for 1000 hours
3. Run *perfect_model_obs* to generate synthetic observations (OSSE)
4. Run *filter* with some adaptive inflation, and 80 members
(divide these into 4 groups if you have worked with the group filter)

Use matlab diagnostics to examine results

Try *plot_ens_time_series*

Select a variable close to the observations and one far away.

Designing Localized Observations for bgrid

1. Run *create_obs_sequence*
 - Enter only 1 observation, 0 values and qc fields
 - Select Radiosonde temperature
 - Vertical coordinate Pressure, 500 hPa
 - Try longitude and latitude 270, 45
 - Time is 0 days, 0 seconds
 - Error variance is 1.0
2. Run *create_fixed_network_seq*
 - Select regularly repeating, 2 times
 - 0 days 0 seconds for initial time
 - 0 days 3600 seconds for period
3. Run *perfect_model_obs*
4. Run *filter*
5. Create innovations: *ncdiff analysis.nc preassim.nc.nc Innov.nc*
6. Use *ncview* to look at copies 1 (mean) and 2 (spread) for fields
Interesting to see how T obs impacts u, v, surface pressure

Selecting Observation Types to Process

Need to specify via name (character string) in namelist:

Type of all observations to be assimilated;

Type of all observations to be evaluated but not assimilated.

(Forward operators are computed and stored in
obs_sequence file).

List of available observation types found in:

assimilation_code/modules/observations/obs_kind_mod.f90.
(see declaration for *obs_type_info*).

Specify in *obs_kind_nml* using names:

&obs_kind_nml

 assimilate_these_obs_types = 'RAW_STATE_VARIABLE'

 evaluate_these_obs_types = 'RAW_STATE_1D_INTEGRAL'

/

Selecting a Set of Observation Definitions

Compile and run the program *preprocess* first.

assimilation_code/modules/observations/DEFAULT_obs_kind_mod.F90 and
observations/forward_operators/DEFAULT_obs_def_mod.F90 are merged with
additional special observation definition files
(in *observations/forward_operators*) to create
assimilation_code/modules/observations/obs_def_mod.f90 and
observations/forward_operators/obs_kind_mod.f90.

See section 21 for more details.

This means to change anything in *obs_def_mod.f90* or
obs_kind_mod.f90, you have to change the DEFAULT files or your
changes will be lost the next time *preprocess* is run.

DART Tutorial Index to Sections

1. Filtering For a One Variable System
2. The DART Directory Tree
3. DART Runtime Control and Documentation
4. How should observations of a state variable impact an unobserved state variable?
Multivariate assimilation.
5. Comprehensive Filtering Theory: Non-Identity Observations and the Joint Phase Space
6. Other Updates for An Observed Variable
7. Some Additional Low-Order Models
8. Dealing with Sampling Error
9. More on Dealing with Error; Inflation
10. Regression and Nonlinear Effects
11. Creating DART Executables
12. Adaptive Inflation
13. Hierarchical Group Filters and Localization
14. Quality control
15. DART Experiments: Control and Design
16. Diagnostic Output
17. Creating Observation Sequences
18. Lost in Phase Space: The Challenge of Not Knowing the Truth
19. DART-Compliant Models and Making Models Compliant
20. Model Parameter Estimation
21. Observation Types and Observing System Design
22. Parallel Algorithm Implementation
23. Location module design (not available)
24. Fixed lag smoother (not available)
25. A simple 1D advection model: Tracer Data Assimilation