DART uses observation sequence files to store information about observations that are available for assimilation.

Default names are:

1. *obs_seq.in*  
   Input to *perfect_model_obs* for OSSEs
2. *obs_seq.out*  
   Input to *filter*, (output from *perfect_model_obs*).
3. *obs_seq.final*  
   Output from *filter*.

These files contain metadata describing observations, and may include a number of related values (for instance, the actual observation, the prior ensemble estimates, etc.).
Each observation can have a prior quality control value. For instance, BUFR files from NCEP contain a prior qc value.

A DART quality control value is added when observation sequences are generated by DART programs. It has metadata *DART quality control*. obs_seq.final generated by filter has following DART qc values:

0. Assimilated
1. Evaluated only
2. Assimilated but posterior forward observation operator(s) failed
3. Evaluated only but posterior forward observation operator(s) failed
4. Not used, prior forward observation operator(s) failed
5. Not used because not selected in obs_kind_nml
6. Not used, failed prior quality control check
7. Not used, violated outlier threshold
DART provides runtime control over what types of observations in an observation sequence file are assimilated by the filter.

This is done in the &obs_kind_nml:

- `assimilate_these_obs_types` is a list of names of observation types to be assimilated.
- `evaluate_these_obs_types` is a list of names of observation types to be evaluated only (withheld observations).

An observation type that is not in either list is ignored.

See `input.nml` in `models/bgrid_solo/work` for an example of an `assimilate_these_obs_types` list.
DART qc values 0 to 3 indicate that the observation was okay
0 and 2: observation was assimilated
  Obs. kind is `assimilate_these_obs_types` in `&obs_kind_nml`
1 and 3: Prior observation ensemble computed, not assimilated
  Obs. kind is `evaluate_these_obs_types` in `&obs_kind_nml`
  This is withholding an observation to be used for validation
2 and 3: one or more posterior forward operators failed
  Cannot use this observation for posterior diagnostics
  Can be used for prior diagnostics

DART qc value 5 indicates observation not used at all.
  Not listed in `&obs_kind_nml`.
  Not used in either prior or posterior diagnostics.
DART qc value 6 indicates that the prior qc value was too large. Threshold set by &quality_control_nml: input_qc_threshold

If prior qc of observation is greater than threshold, then...
  Observation is not assimilated.
  Not used in either prior or posterior diagnostics.

NOTE: BUFR qc values larger than 3 means observation is suspect. Most people assimilating observations from BUFR use an input_qc_threshold of 3 which is the default value in DART input.nml’s.
DART qc value 7 indicates outlier threshold exceeded

\[ \text{Expected}(\text{prior mean - observation}) = \sqrt{\sigma_{\text{prior}}^2 + \sigma_{\text{obs}}^2} . \]

Reject if (prior_mean - observation) > T times expected value. T is set by \textit{outlier_threshold} in \&\textit{quality_control_nml}. \textit{outlier_threshold} < 0 means no outlier check.
Outlier threshold quality control

Designed to discard observations that are inconsistent with prior.

Low-order models have the `outlier_threshold` check turned off by default, set to -1.

Large models have `outlier_threshold` set to 3 standard deviations.

Setup a successful lorenz_96 or lorenz_63 assimilation case. Set `outlier_threshold` to 2.0, or 1.5. Examine what happens to assimilation quality.

Outlier threshold qc has a significant impact when using BUFR observations.
1. Filtering For a One Variable System
2. The DART Directory Tree
3. DART Runtime Control and Documentation
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