# Observations Database data import version 2

### Tim Parker-Nance, Nimbus Services

The setup and import of data logs in version 1 is very tedious, rigorous and not resilient to change. There are also missing features (like R integration, automation) that could make the technician/scientist’s work easier. The version 2 data import endeavours to resolve these problems.

Some perspective first, SAEON has number sites, each with a number of stations, each with a number of instruments, each with a number of sensors, each that measure a phenomenon and possibly a number of offerings. So for example we might have a Vantage Pro2 Weather Station with sensors for temperature, wind direction, wind speed. These are sampled every minute, hour and day. We also get average, minimum and maximum for the sample period. The samples are recorded in a Comma Separate Value (CSV), Fixed Column Width (Flat) or Proprietary formats.

### Import Process

So what is the process we go through to get from a log file to valid observations in the database? The process is the following:

* Data file storage: Uploading data file for safe storage and repeatable reads
* Data file read: Read data file to in-memory tabular format persisted as XML
* Column mapping: Remove columns that are not valid for the import period
* R pre-process: Pass the tabular data to R for cleaning and quality assurance.
* Transformations: Transform the observed values in the tabular data from the DataSourceTransformations, possibly marking value with a comment due to quality assurance problems.
* R post-process: Pass the tabular to R for quality assurance.
* Quality assurance: Final checks to mark dataset as ok or problematic.
* Import: If passes Quality assurance checks create observations in the database from the tabular data.

### Repeatability

What is repeatability? If we find a problem with an import later we can change a transformation or add a new transformation and then redo the import. The problem is that log files can go missing so without it a re-import is impossible. The old system uses the DataLog for repeatability, although it is not required. The problem is that SQL is notably inept at handling unstructured data. The DataLog is a manifestation of such unstructured data. It basically duplicates each observation for repeatability. This will have huge performance implications in future when the database gets huge. So what are the options? SQL Server can store the unstructured data as Blobs. The problem is that these Blobs are stored in the database, and even if they are hardly accessed they have performance implications when the database grows. A better solution is a FileStream, a special Blob that lives in the operating system and not the database, but is managed by SQL Server. So it can’t be deleted by a user, it can only be deleted via the database. Also the FileStream Blobs are backed-up in the database backup, so there is no need for a second backup of the log files. With data files, the new DataLog (XML) and when available, the AcSys log files stored as FileStream Bolbs repeatability can be achieved without impacting database performance.

### Resilience

The current DataSource/DataSchema mechanism is too rigid, rigorous and not resilient to change. Let’s ignore the old system and describe how it should work. Ideally this should be a setup once process with the system being more intelligent in handling changes. Sensors break, malfunction, etc. The incidents should not require setup changes.