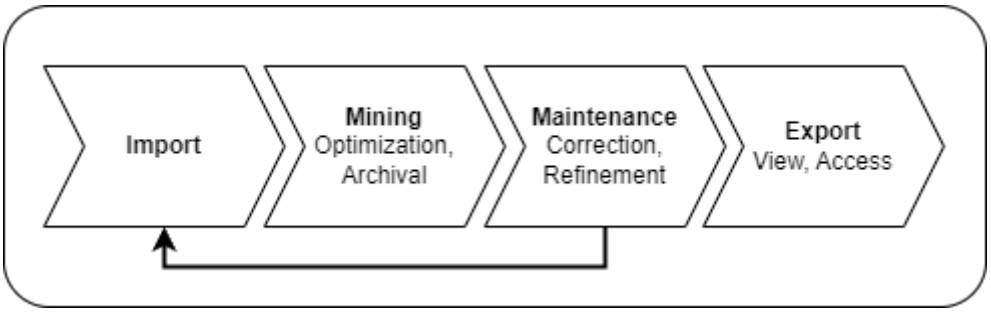
Data Lifecycle

The XINA structs mnemonic data lifecycle involves four primary phases:



Source Files

Each pipe maintains a set of **source files**, containing all data imported into XINA for that pipe.

The primary type of source files are **archive source files**. Archive files are considered the **definitive record of source data for a range of time for a single pipe**. These are stored in the XINA xbin binary file format. These are imported directly with the STRUCT ARCHIVE IMPORT action. Archive files are **mined** through the XINA Structs Mine task into XINA databases in order to be viewed in the XINA client, and are used to generate export packages.

Alternatively, an pipe may use **buffer source files**. Buffer files may be imported in a variety of data formats and are not subject to the same strict requirements as archive files. These may are imported directly with the STRUCT BUFFER IMPORT action. Mnemonic data from buffer files is loaded into a temporary buffer database for immediate viewing in the XINA client. Buffer files are **archived** (merged and converted into archive files) through the XINA Structs Archive task, which can be run manually or configured to run in regular intervals. *This is the recommended approach for importing mnemonic data when getting started with XINA Structs.*

Data Flow

In general, there are three supported approaches for pipe data flow: **buffer** import, **variable time archive** import, and **fixed time archive** import. While a single pipe can only support one workflow, a model may combine multiple workflows using multiple pipe.

Buffer Import

The buffer import workflow is the most flexible mnemonic import method. Buffer files do not need to adhere to strict requirements (aside from conforming to standard accepted file formats). Buffer files for a given pipe may have duplicated data, overlapping data, and can introduce new mnemonic definitions on demand.

Buffer files are imported with the STRUCT BUFFER IMPORT action. This invokes three effects:

- the raw buffer file is parsed, validated, and stored in the model pipe **mnemonic buffer file database**
- new **mnemonic definitions** are created for any unrecognized mnemonic labels
- data is added to the **mnemonic buffer database** for the associated pipe

No additional data processing occurs as part of this step. XINA models utilizing buffer source files must implement routine execution of the `STRUCT_BUFFER_ARCHIVE` asynchronous task (typically every hour) to merge the files into archive files in a fixed-time archive format, which can then be processed by `STRUCT_ARCHIVE_MINE` tasks to fully process data into model standard databases.

Pros

- minimal client side configuration required to get started
- allows smaller, faster file uploads to view data close to real-time
- flexible and responsive to changing environments, mnemonics, requirements

Cons

- performance is worse than client side aggregation
- not recommended above 1k total data points per second

Struct Archive Task

The XINA Struct Archive task merges and compresses buffer files into archive files. This step is required to resolve any data discrepancies and ensure data is preserved in accordance with the requirements of archive files. The task performs the following steps:

- load all unprocessed files from the buffer file database
- for each time ranges affected by unprocessed files
  - process each file into processed format
  - load any existing processed files in those time ranges
  - merge data from all processed files for time range into single archive file
  - upload newly processed buffer files
  - delete unprocessed buffer files
  - upload merged archive file
  - run mining task on merged archive file
  - delete any mnemonic data already present for time range
  - import mnemonic data generated by mining task

Direct Archive Import

Archive files are imported directly with the STRUCT ARCHIVE IMPORT action.

Pros

- much higher performance ceiling than server side aggregation
- stringent validation ensures data conforms to standard

Cons

- more complex initial setup
- mnemonic definitions must be pre-defined and cannot be added on-the-fly
- mnemonic definitions need coordination between client and server
- changes are more complex and likely involve human interaction

Fixed-Time Archive Import

With fixed-time archive import each archive has a fixed time range. This is a recommended solution for projects which generate a persistent data stream (for example, data sources piped through a FEDS server).

Variable-Time Archive Import

With variable-time archive import each archive specifies a custom time range. This is a recommended solution for projects which generate their own archival equivalent (for example, outputting a discrete data set after running a script). Because the time ranges are determined by the source data, it is recommended to generate interval events matching each file as a time range reference.

Source File Formats

Currently there are two natively supported general purpose formats, one using the codes `csv` / `tsv` (full documentation here), and a binary format using the code `xbin` (full documentation here) Additional formats will be added in the future, and custom project-specific formats may be added as needed.

Assumptions and Limitations

Each archive source file is considered the **single source of truth for all mnemonics, instants, and intervals for it's associated pipe for its time range**. This has the following implications:

Archive files with the same pipe cannot contain overlapping time ranges. If an import operation is performed with a file violating this constraint the operation will fail and return an error.

Within a single model, each mnemonic may only come from a single pipe. Because mnemonics are not necessarily strictly associated with models, and the source may vary between models, this cannot be verified on import and must be verified on the client prior to importing data.

Revision #49
Created 28 July 2022 15:55:25 by Nick Dobson
Updated 27 August 2024 14:08:57 by Nick Dobson