Build All Data

(Read raw files and save processed data to archive)

Start

Select data within new date range from archive in memory. Save selected data to working files and new date range to config file.

Work with selected Dates

Archive Rebuild?

Read Working Files

New Date Selection

yes

no

no

Within Last Selection?

Read Archive Files

Rebuilt?

no

yes

yes

no

Rebuilt?

yes

yes

no

Read Working Files

Startup

Select data within new date range from archive in memory. Save selected data to working files and new date range to config file.

Read Working Files

New Date Selection

no

Within Last Selection?

Read Filtered Archive Files

Rebuilt?

no

yes

yes

no

Rebuilt?

yes

yes

no

Read Working Files

Read RAW from archive and:

Apply filters to delete bad data.

Two kinds so far:

* Global: met\_filter()
* Manual: apply\_edits()

Save files to “.\archive\filtered”

Inventory, Build and Save:

Gets TIMESTAMP of first and last line of all raw data files in a folder with data type determined by filename and first character in file.

For each data type: Merges all files together (concat\_df) and returns a data frame for each type. Reads EC files twice to maintain format while enabling gap analysis and appropriate merge with MET and RAD data.

Saves four data files to “.\datastore\proj\archive\raw”

and inventory files to “.\datastore\proj”

Archive Rebuild?

yes

no

Archive Rebuild

or

Filter?

yes

no

Return

From Startup

Impute Values

no

yes

Rebuilt?

yes

no

Select data within new date range from archive in memory. Save selected data to working files and new date range to config file.

Work with selected Dates

Read Filtered Archive Files

Read Working Files

Read RAW from archive and:

Apply filters to delete bad data.

Two kinds so far:

* Global: met\_filter()
* Manual: apply\_edits()

Save files to “.\archive\filtered”

Gap Analysis?

yes

no

Build All Data

Use inventory to select

yes

yes

no

For each data type, build a dictionary of data gaps:

* Pad missing time stamps with blank rows.
* For each column of data, count from beginning of gap to end and record length indexed by starting TIMESTAMP.

Gap Plots?

yes

no

For each data type with a gap, make a plot if possible.

Impute\_master(): So far just working with MET data but trying to build something generalizable:

* First, good place for plots to see data state after filtering and padding
* Gets daily and monthly means.
* For each column, fill padded rows with gap length so gap filling can be length dependent.

From Startup

Impute Values

no

yes

Rebuilt?

yes

no

Select data within new date range from archive in memory. Save selected data to working files and new date range to config file.

Work with selected Dates

Read Filtered Archive Files

Read Working Files

Read RAW from archive and:

Apply filters to delete bad data.

Two kinds so far:

* Global: met\_filter()
* Manual: apply\_edits()

Save files to “.\archive\filtered”

Gap Analysis?

yes

no

Build All Data

Use inventory to select

yes

yes

no

For each data type, build a dictionary of data gaps:

* Pad missing time stamps with blank rows.
* For each column of data, count from beginning of gap to end and record length indexed by starting TIMESTAMP.

Gap Plots?

yes

no

For each data type with a gap, make a plot if possible.

Impute\_master(): So far just working with MET data but trying to build something generalizable:

* First, good place for plots to see data state after filtering and padding
* Gets daily and monthly means.
* For each column, fill padded rows with gap length so gap filling can be length dependent.

Automatic process to apply edits that are identified and entered manually.

Overview: Manually create file listing bad data then read file with code to set to missing (similar to a mask).

Begin with pre-filtered data (after range and limit checks).

Open csv file of pre-filtered data with excel.

Plot related variables.

Identify bad values “by eye.”

Cut and paste bad values into new worksheet in deletion workbook file.

Copy-and-paste corresponding dates.

I:\ALL\TampaET\FieldFiles\python\datastore\DeadRiver\_02\archive

**\Met\_Deletions.xlsb**:

Excel workbook with worksheets that identify bad data.

Worksheet: CS215

List of Air Temp’s and RH’s to delete

Columns:

* TIMESTAMP
* Deleted Temp
* Deleted RH

Worksheet: KPSI

List of Water Levels and WL Temp’s to delete.

Columns:

* TIMESTAMP
* Deleted WL
* Deleted WL\_temp

Output individual worksheets as csv files (or check reading worksheets in pandas)

Re-run program and apply deletions after limit checks.

* apply\_edits(path\_out, df)
* input\_and\_merge\_edits(df, df\_deletions, delete\_list)