Multi-Function File Utility – GUI Version

Multi-Function File Utility – GUI Version

=========================================

Version: 1.0

Last Updated: June 14, 2025

Author: Marc DeJong

Environment: Python 3.12+, Tkinter, Dask, Pandas

Overview

--------

This utility provides a modular, GUI-based system for performing advanced operations on structured CSV/TXT data files. Designed for both power users and non-programmers, the system supports low-memory stream mode and full-file parallelized Dask mode, automatically selected based on function requirements.

## The main entry point is:

python 0x\_File\_Util\_Caller\_gui.py

Supported Functions (Fully GUI-Enabled)

----------------------------------------

| Menu # | Function Name | Description |

|--------|----------------------------------------|-------------|

| 1 | filter\_omit | Remove records that match one or more filter rules |

| 2 | filter\_select | Retain only records that match filter rules |

| 3 | replace\_rec\_contents | Replace values in specified fields based on rules |

| 4 | add\_rec\_stub\_(fixed) | Add a new column with a fixed value |

| 5 | add\_rec\_stub\_(var\_from\_rec\_contents) | Add a new column with values derived from existing fields |

| 6 | delete\_rec\_by\_condition | Remove records that match deletion rules |

| 7 | sort\_records | Sort records by one or more fields |

| 8 | dedupe\_records | Remove or extract duplicate records |

| 9 | split\_file\_by\_condition | Split file by match/no-match on a single field |

| 10 | split\_by\_composite\_condition | Split file by multiple matching fields (composite key) |

| 11 | concatenate\_files | Combine two or more files vertically |

| 12 | merge\_by\_key | Retain or exclude records based on key file matches |

| 13 | compare\_two\_files\_gui | Standalone GUI to compare two delimited files |

Features

--------

- Header-aware: Prompts for header presence and uses named fields when available.

- Flexible input: Accepts CSV/TXT files with any delimiter.

- Error-tolerant: GUI error dialogs and pre-validation prevent crashes.

## - Dual processing modes:

- Stream-based mode for memory-efficient line-by-line operations

- Dask mode for high-performance full file operations with sorting, deduping, and composite filtering

- Modular architecture: All functions are in independent modules and can be extended or swapped out.

- Preview and logging: Console output summaries and GUI confirmations for key operations.

Folder & Module Structure

--------------------------

| File | Purpose |

|------------------------------|---------|

| 0x\_File\_Util\_Caller\_gui.py | Main controller GUI – load file, select function(s), route execution |

| 1x\_User\_Interface.py | Not used in GUI version |

| 2x\_File\_Loader\_gui.py | File input/output + delimiter detection |

| 3x\_Filters.py, filters\_ui\_3x\_gui.py | Filter logic and GUI prompts |

| 4x\_Replacer\_gui.py, 4x\_Replacer\_UI\_gui.py | Replace logic and config GUI |

| 5x\_Add\_Stub\_gui.py, 5x\_Add\_Stub\_UI\_gui.py | Add stub logic and GUI config |

| 6x\_Deleter.py | Stream-based deletion logic |

| 7x\_Sorter.py | Sort function for Dask-based processing |

| 8x\_Deduper.py | Remove or retain duplicates (by field) |

| 9x\_Splitter.py, splitter\_9x\_gui.py, splitter\_9x\_composite\_gui.py | Standard and composite-based file splitting |

| 10x\_Merger.py, merger\_gui\_10x.py | Merge by key and concatenate logic |

| 11x\_Utils.py | Shared helpers for parsing and validation |

| Two\_File\_Compare\_gui.py | Compare two files – write out compare report |

How to Run

-----------

1. Ensure all modules are in the same folder.

## 2. Run the main GUI:

python 0x\_File\_Util\_Caller\_gui.py

## 3. Use the GUI to:

- Load a file

- Check header presence

- Choose one or more operations

- Configure each function through guided prompts

## 🆕 Standalone Utility: File Compare (GUI)

- - - -

This standalone script allows comparison of two delimited files, showing unmatched records with detailed character-level differences.

- Supports files with or without headers

- Allows specific columns to be excluded from comparison

- GUI-based selection of files and fields

- Generates `COMPARE\_RESULTS.txt` in the same folder

- - - -

Requirements

-------------

- Python 3.12+

- Tkinter (bundled with standard Python)

- pandas, dask

pip install pandas dask

---

### 🔄 Recent Updates

\*\*2025-06-17\*\*

- Added new standalone compare utility: `Two\_File\_Compare\_gui.py`

- Updated GUI documentation and menu table with function #13

Future Enhancements (Phase 2 Ideas)

-----------------------------------

- GUI-based config save/load for batch runs

- Dry-run preview mode for all functions

- Full log/summary report with timestamps

- Optional output directory selection

- Fuzzy column matching in merge-by-key

- More robust numeric/date filter enhancements