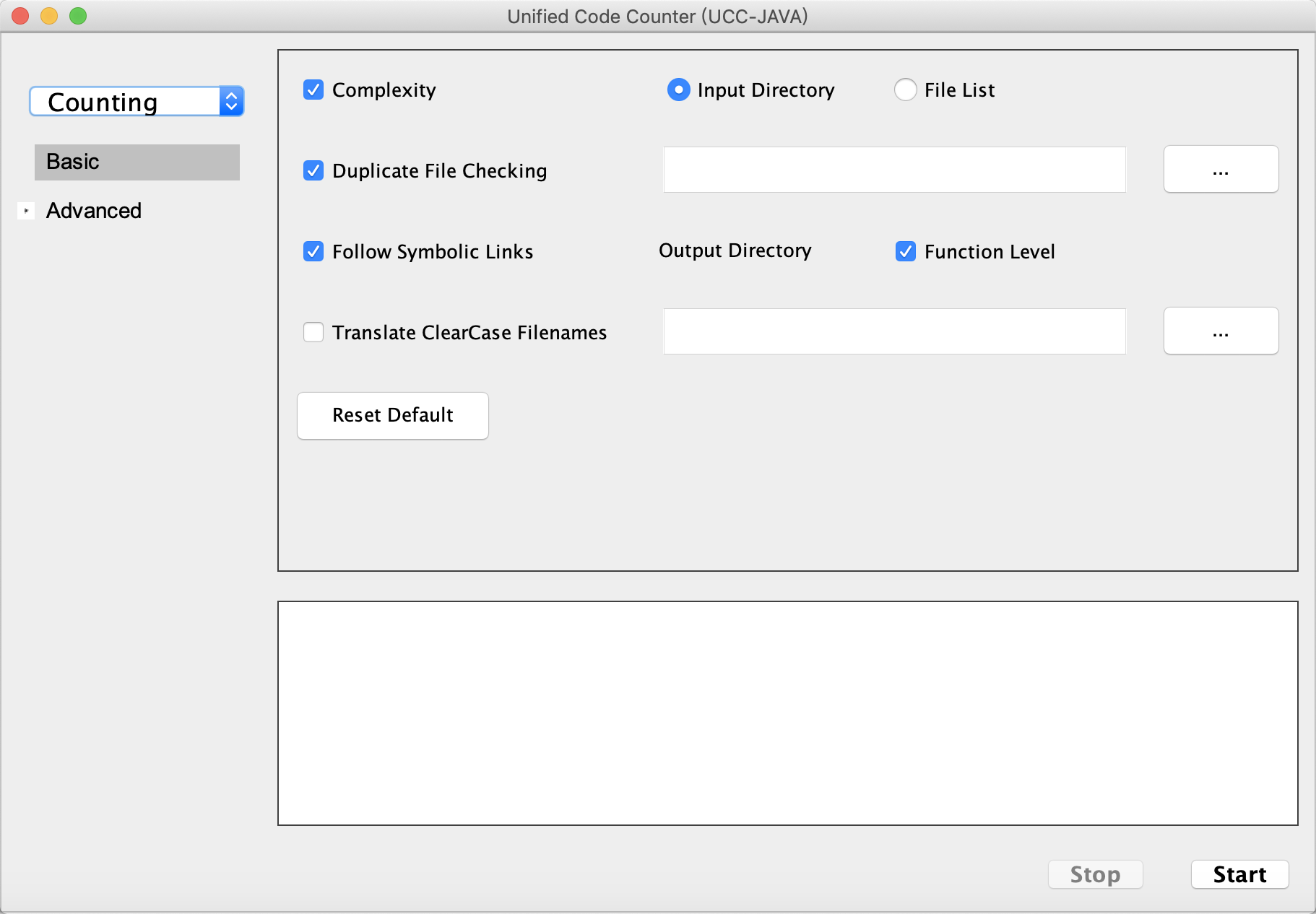
**UCC Java GUI USER MANUAL**

****

**Basic panel:**

1. **Complexity:**

Process language keywords or report complexity metrics. Not using this switch will reduce the processing time.

1. **Duplicate file checking**

Search for duplicate files. Any duplicates will be reported as unique files. This increase processing time.

1. **Follow symbolic links**

Follow symbolic links on Unix based systems. Do not skip Unix symbolic links and multiple counting of same files as links. This enable following symbolic links to directories and counting of links to files.

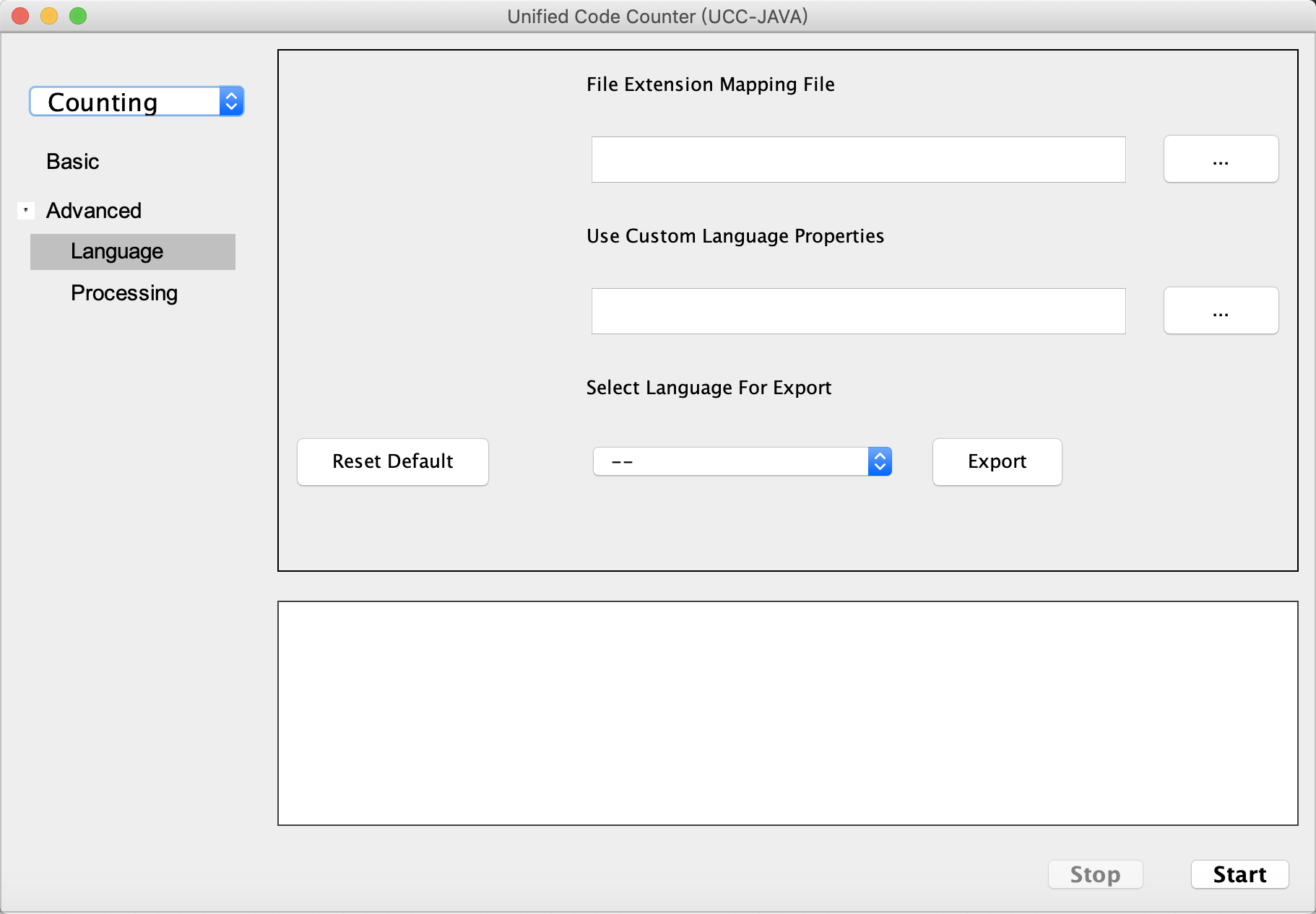
1. **Translate ClearCase filenames**

Support handling ClearCase filenames. The ClearCase application appends version information to the filename, starting from ‘@@’. This option requires the UCC-J tool to handle the original filename instead of the ClearCase-modified filename by stripping off the ‘@@’ and any characters after that and before the extension.

1. **Function Level**

Counting these metrics on function in addition to file level.

**Advanced**

**Language panel**

1. **File extension mapping file**

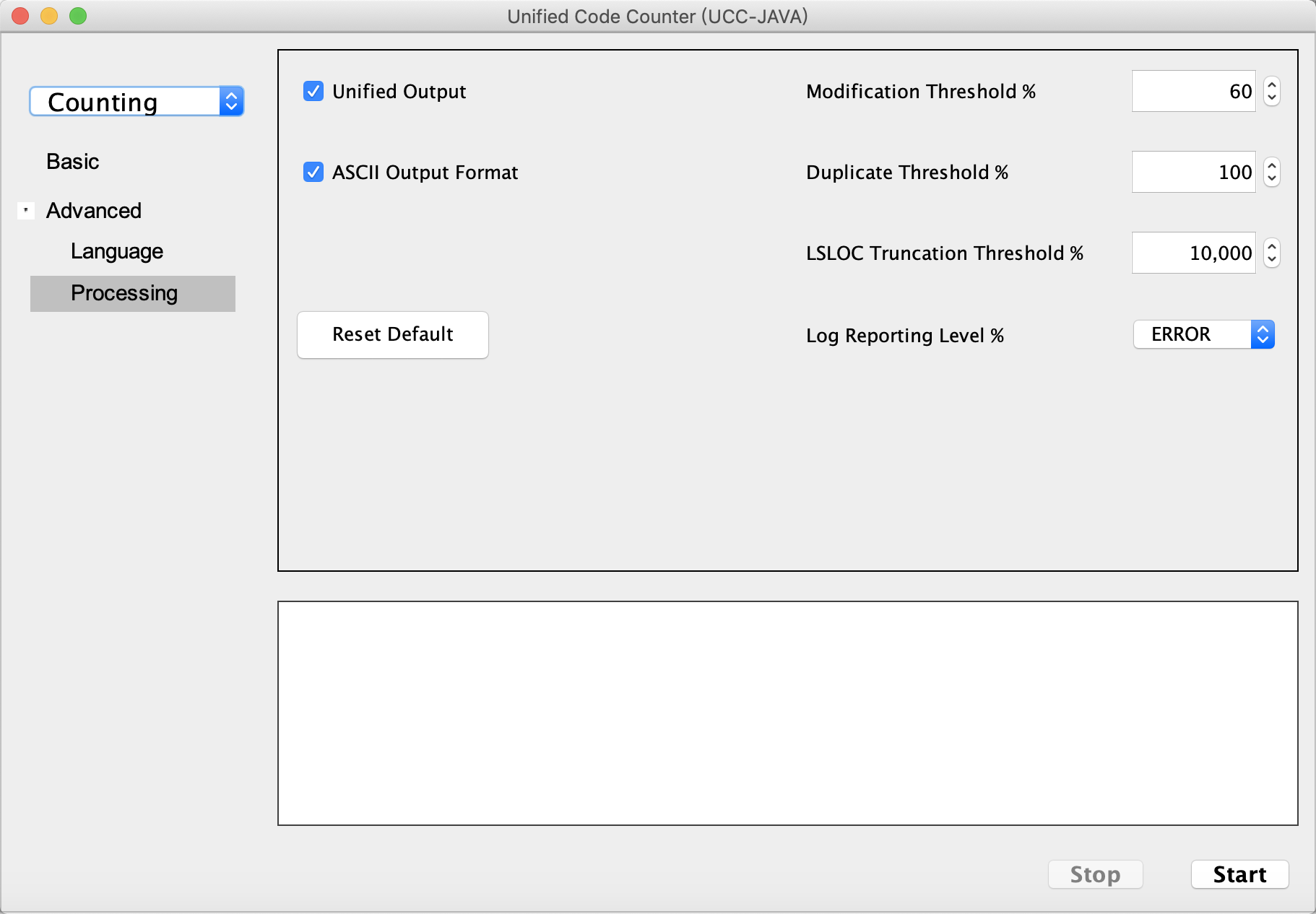
Allows user to specify the name of a file that contains languages and extensions to map to that language counter. This allows the user to include non-default extensions or remove default extensions.

1. **Use custom language properties**

Imports specified file containing language properties for a custom language. The custom language is included when code counter is run on a given directory or a file list.

1. **Select language for export**

Exports language properties for the specified language to a text file. The output text file is stored in the directory specified with output directory option or defaults to current working directory. Output file name will be *[language\_name]LanguageProperties.txt*.

**Processing panel:**

1. **Unified Output**

Directs the UCC-J to print counting results to a single unified language file name *TOTAL\_outfile.csv*.

If the radio button is not selected, a report is produced for each language and is named *<Lang>\_outfile.csv*, which is an Excel format.

1. **ASCII Output Format**

Generate reports in text format (.txt). The default report format is .csv, which directly opens into MS Excel.

1. **Modification Threshold**

Specifies the modification threshold.

The user is given the ability to tailor how the UCC-J determines if a SLOC has been modified using this function. The number is the modification threshold and can be any number between 0 and 100, with the default being 60. The modification threshold specifies a percentage; i.e. 60 indicates a modification threshold of 60%. If two SLOC have the percentage of common characters equal or higher than the specified threshold, as compared over the length of the longest line, they are matched and counted as modified. Otherwise, they are counted as one SLOC deleted and one SLOC added. In this example, using 60, if 60% of the characters of the longest line match with the compared line, the lines are considered modified in Baseline B. If less than 60% of the characters in the longest line match with the compared line, Baseline A counts one SLOC deleted, and Baseline B counts one SLOC added. If the compared lines are exactly the same, the lines are counted as unmodified.

1. **Duplicate Threshold**

Specifies the threshold percentage for identical SLOC when comparing two files within a baseline for one file to be considered a duplicate of the other.

This function specifies the maximum percent match allowed between the SLOC in two files of the same name within a baseline to be considered duplicates. Blank lines and comments are not considered. By default, this threshold is 60. Valid values are numbers between 1 and 100. The threshold corresponds to the percentage of SLOC which may be the same when two files are compared in order to be considered duplicates. For example, 20 would mean that a file is a duplicate of another file if 20% or more of the SLOC are same.

Files must be in the same baseline, but not necessarily in the same directory, in order to be duplicates. The files do not need to have the same name; however, if the file names are different, they have to be exactly the same, including comments and blank lines, to be identified as duplicates. Files with the same name, but not in the same subdirectory, are considered duplicates if the code is identical, even if there are differences in the comments and/or blank lines.

Duplicate files are counted and reported separately. One purpose for this command is to isolate SLOC counts for files which did not require development but were duplicated for a variety of reasons which could include for configuration management purposes, or were computer generated.

1. **LSLOC Truncation Threshold**

The truncation threshold specifies the maximum number of characters allowed in a logical SLOC. Additional characters will be truncated. The default value is 10,000. If 0 is specified, no truncation is done. Performance can be significantly degraded if truncation is too high.

1. **Log Reporting Level**

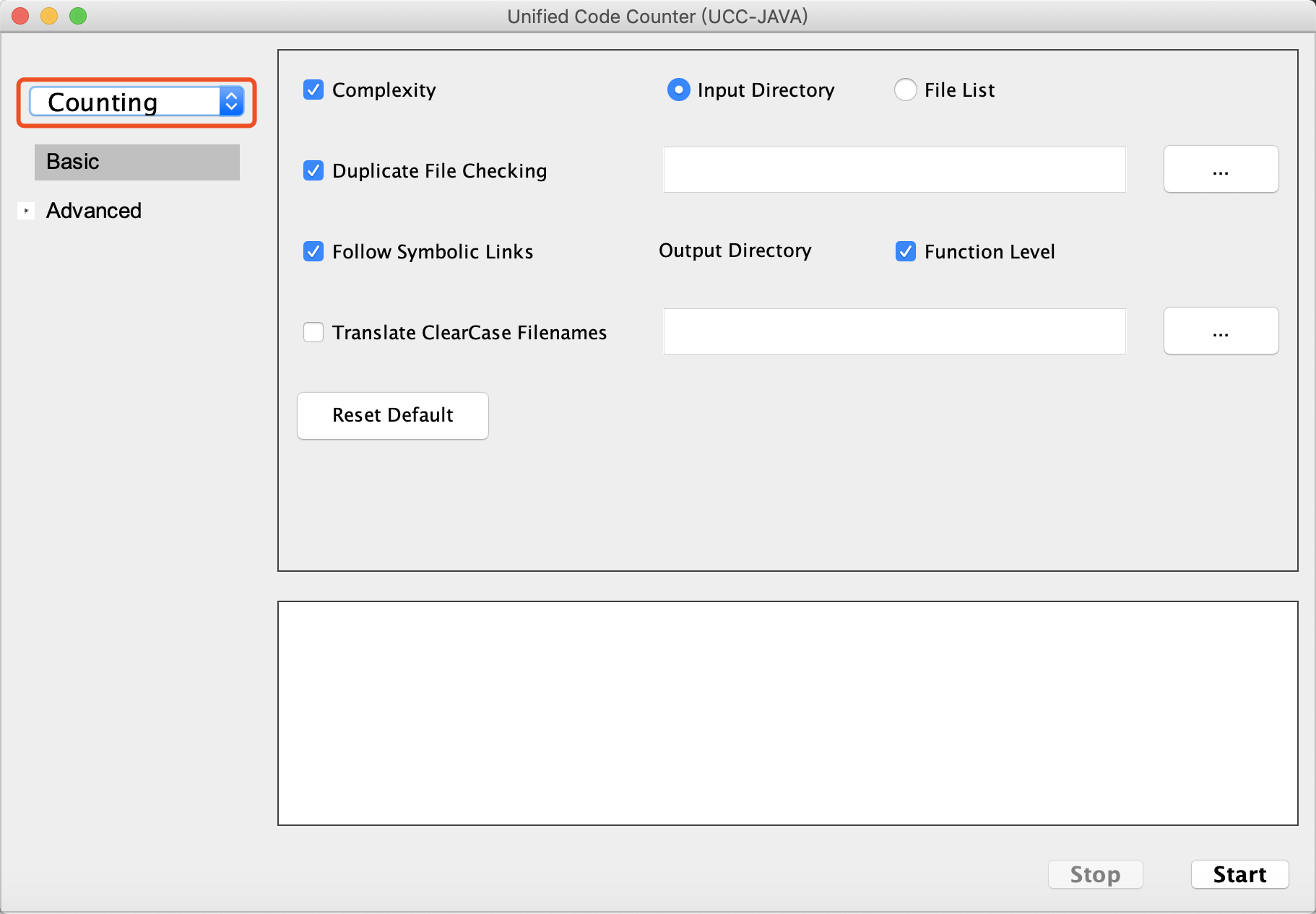
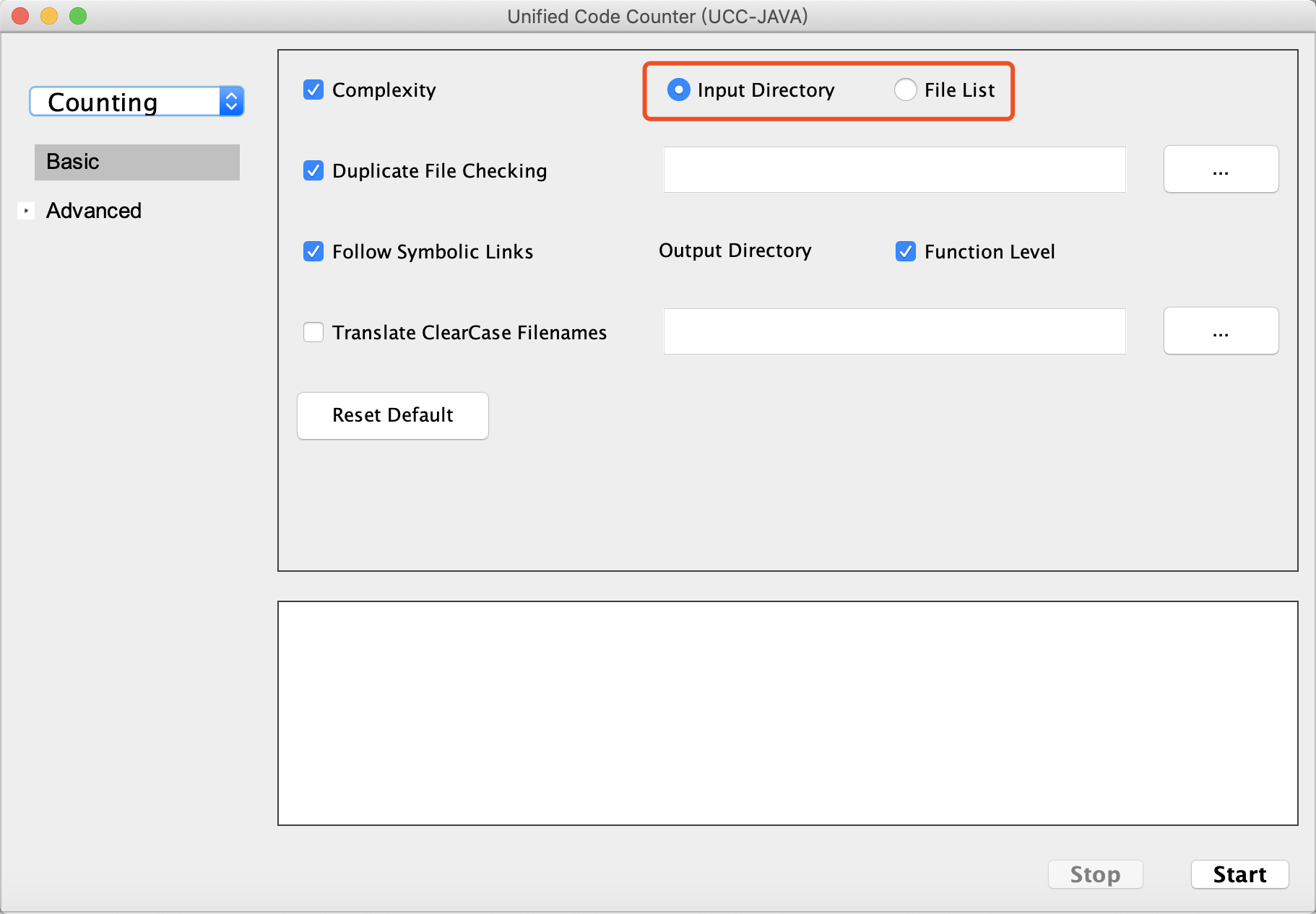
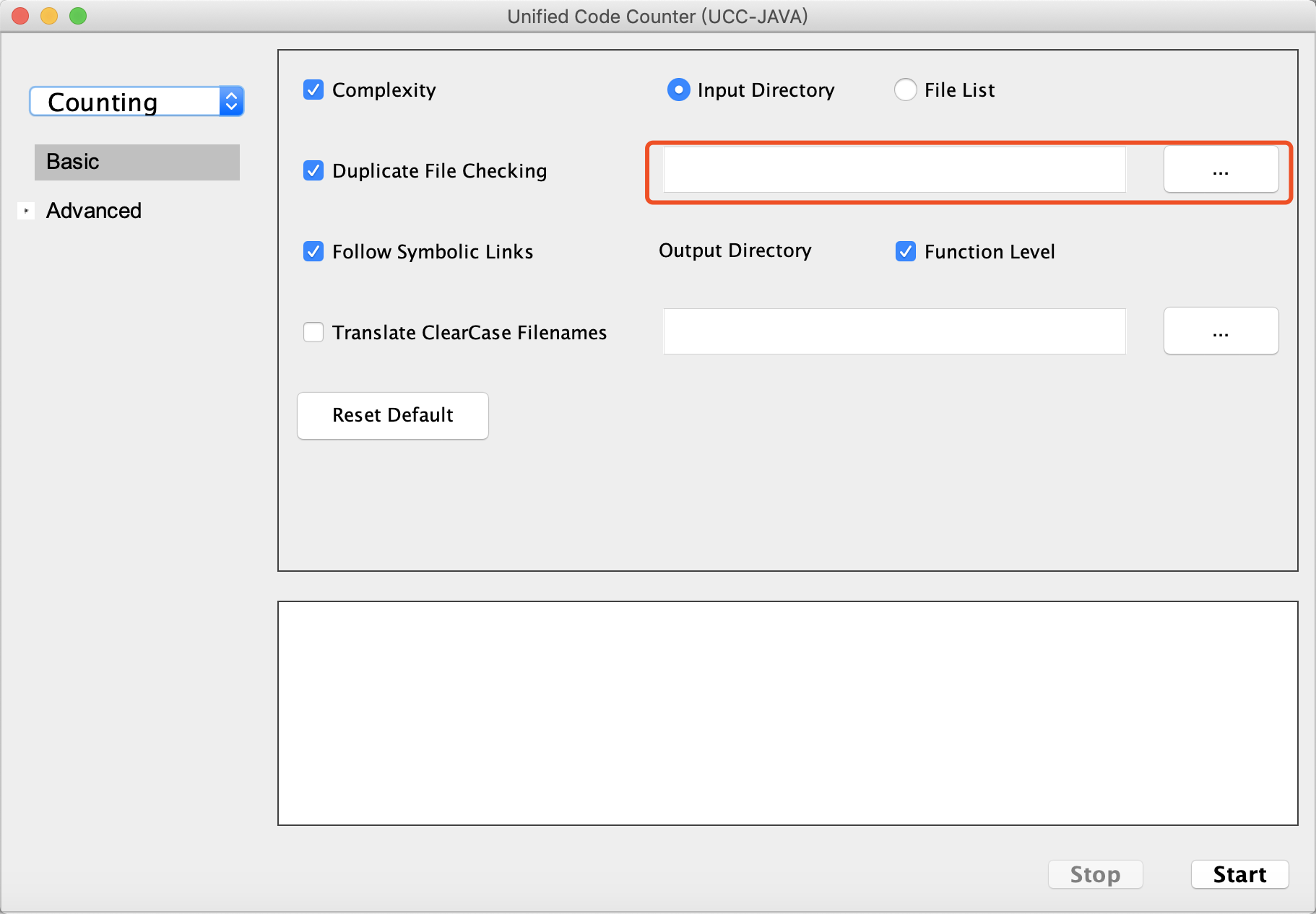
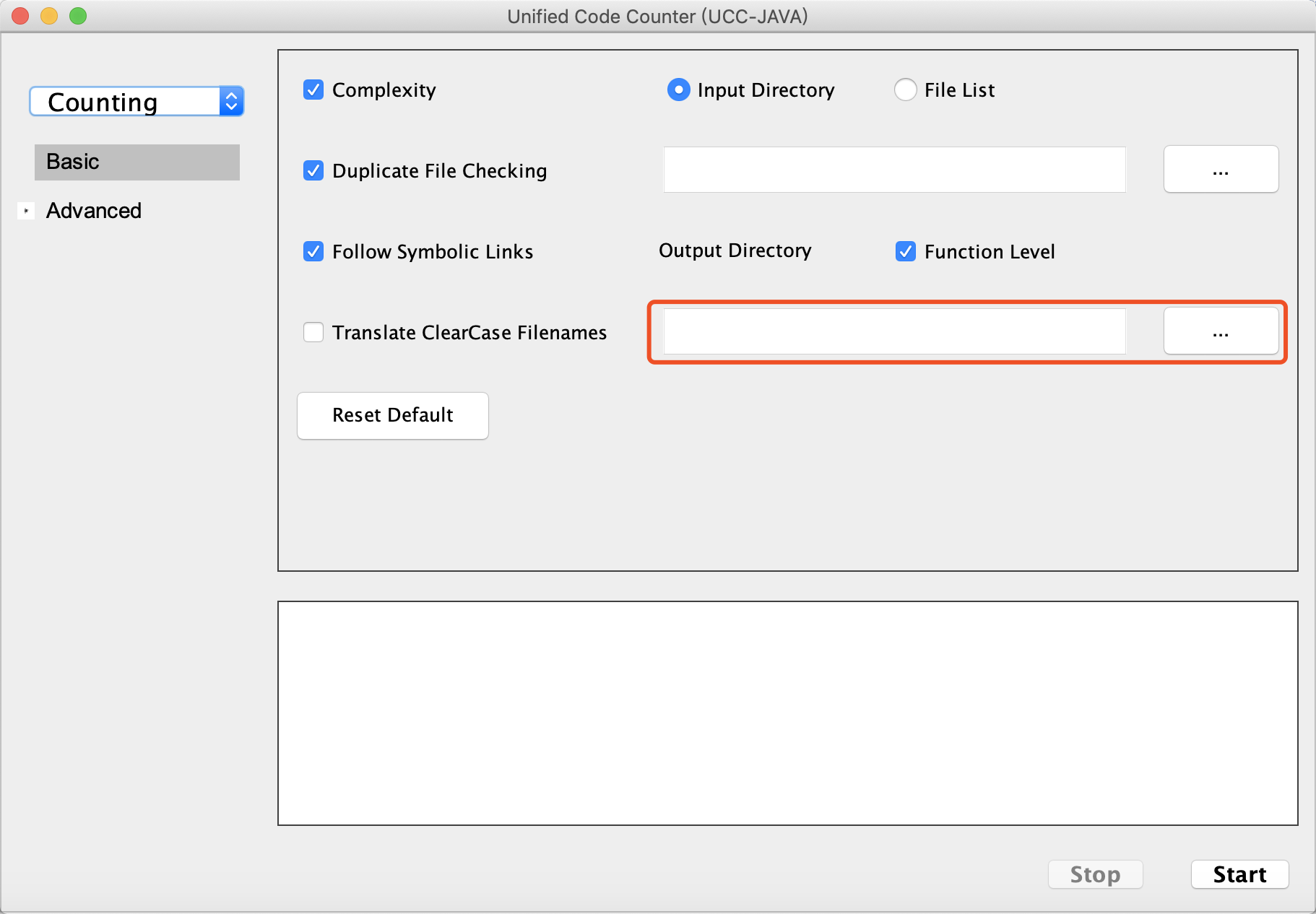
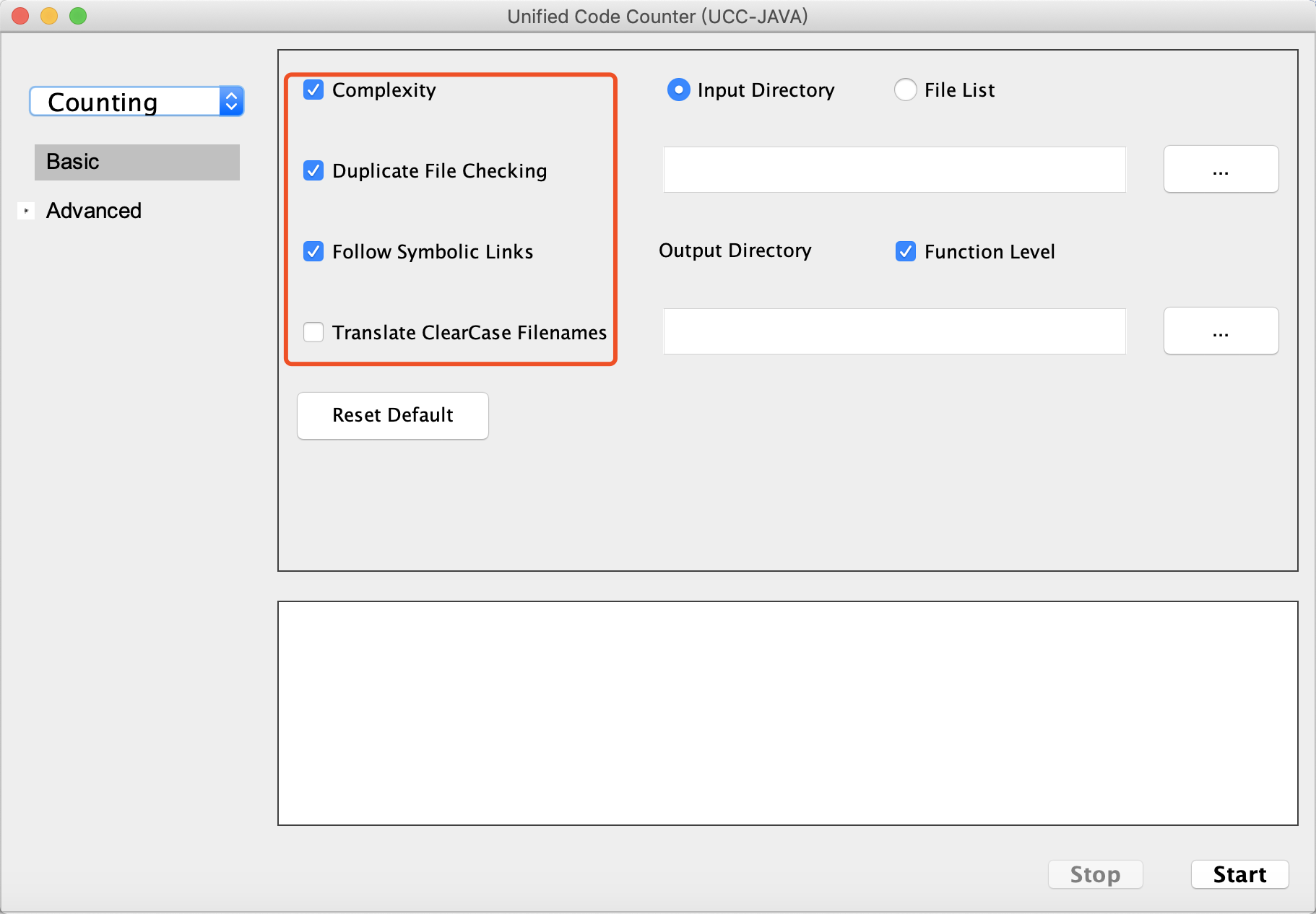
Enables the UCC-J debugger. Outputs are directed to the *<time\_stamp>\_ucc.log* file which will appear in the same folder as the UCC-J executable JAR by default if an output directory is not specified.

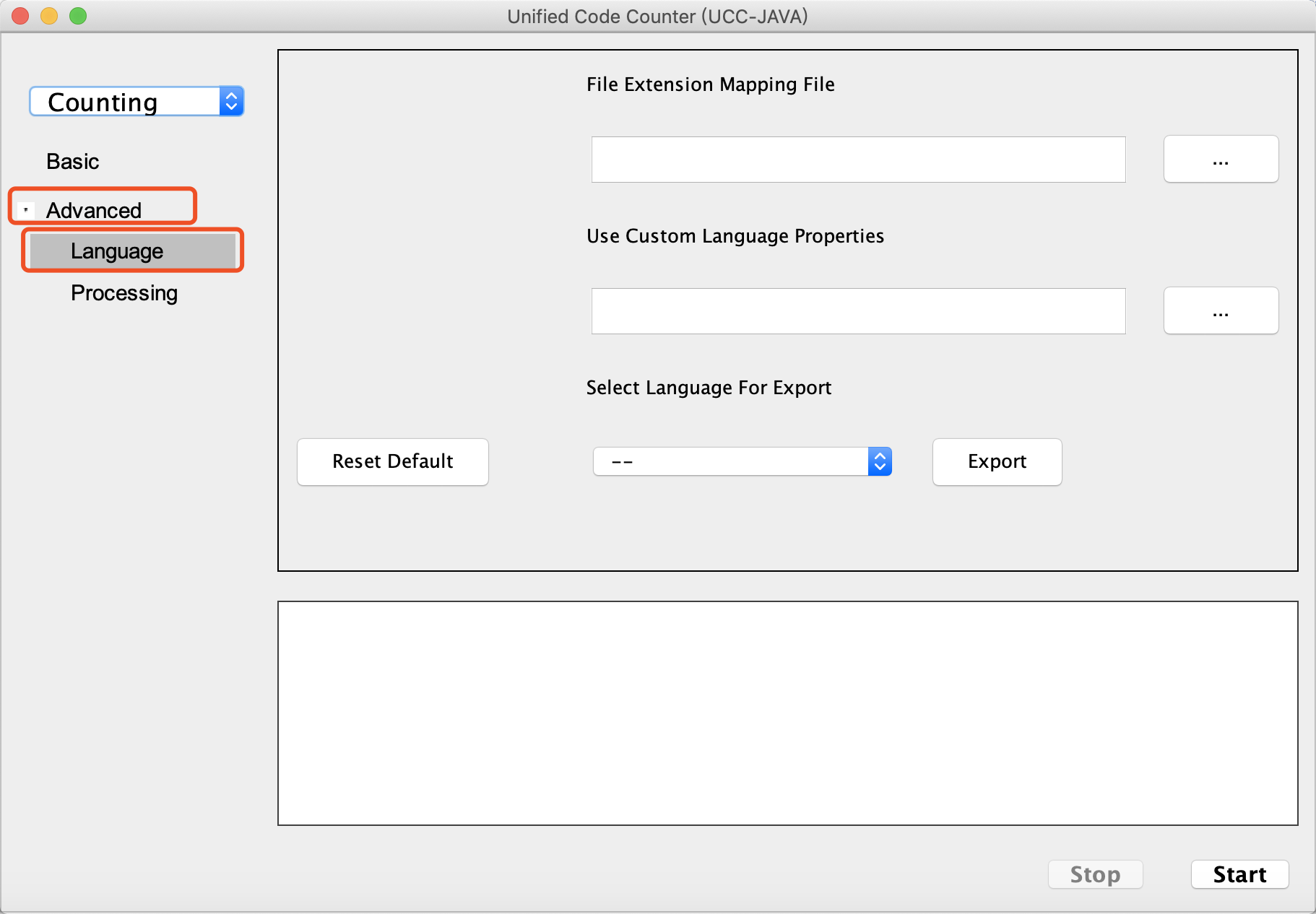
Valid levels are FATAL, ERROR, WARN, INFO, DEBUG, or TRACE.

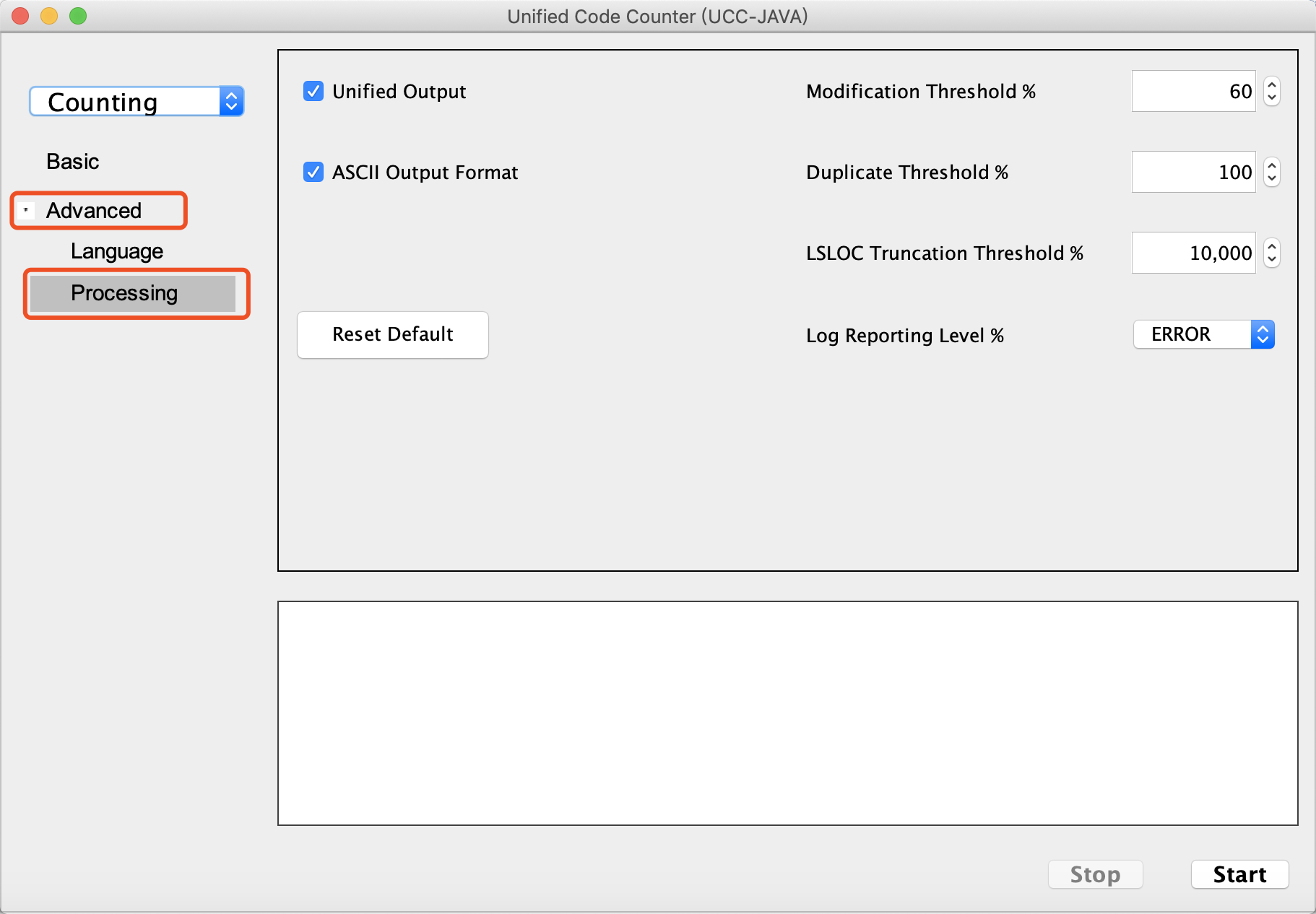
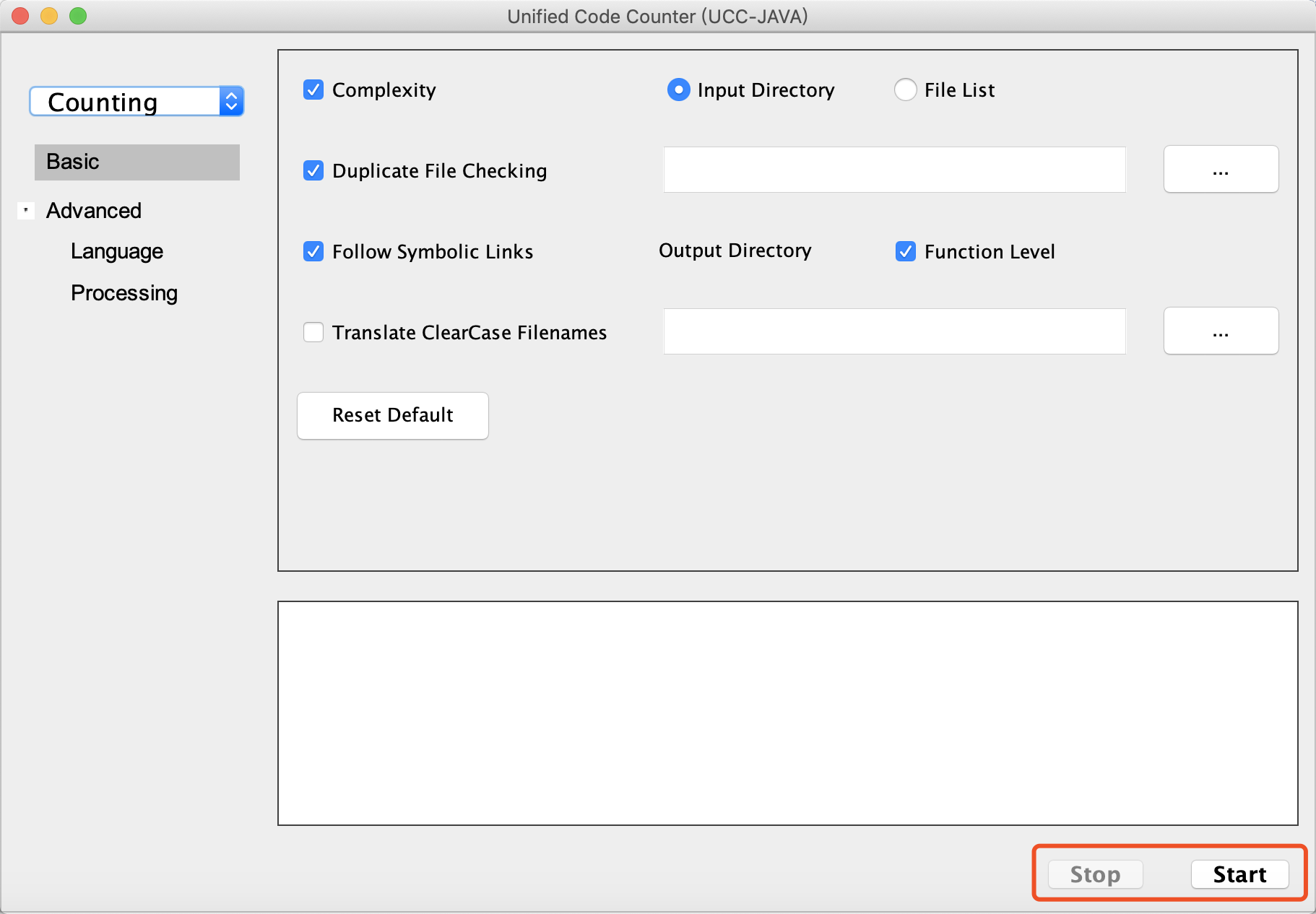
By default, the application is set to the ERROR level.

All levels under the specified level will be logged. For example, if the level is set to INFO, any log outputs with INFO, WARN, ERROR, or FATAL level would be outputted and log statements with DEBUG or TRACE would be ignored.

**Counting Example:**

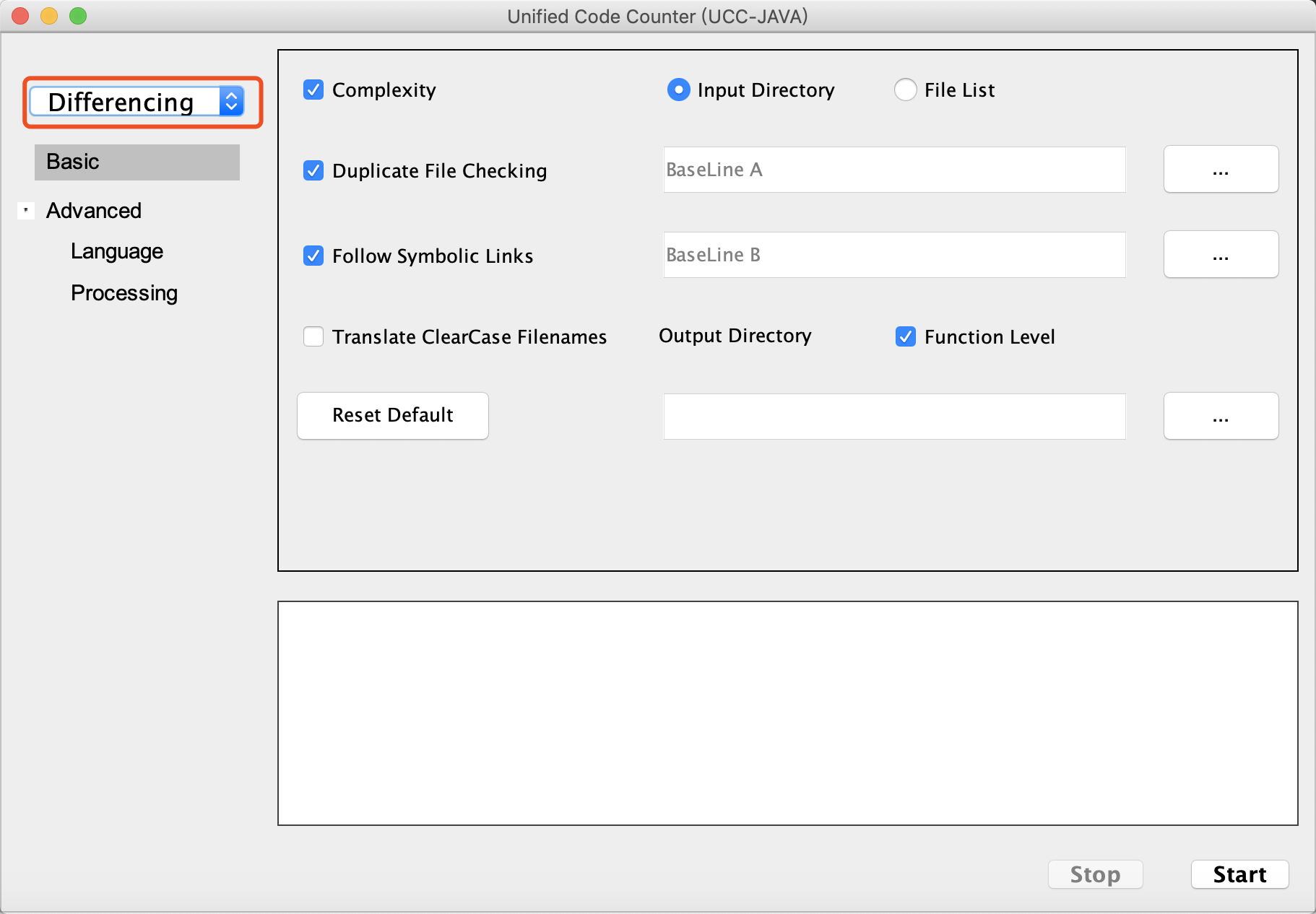
1. **Set GUI function to “Counting”**
2. Set Input type to “directory” or “filelist”
3. Browse directory or file on local computer
4. Set output directory for report files
5. Customize the basic settings (optional)
6. Customize the language setting in advanced tab (optional)

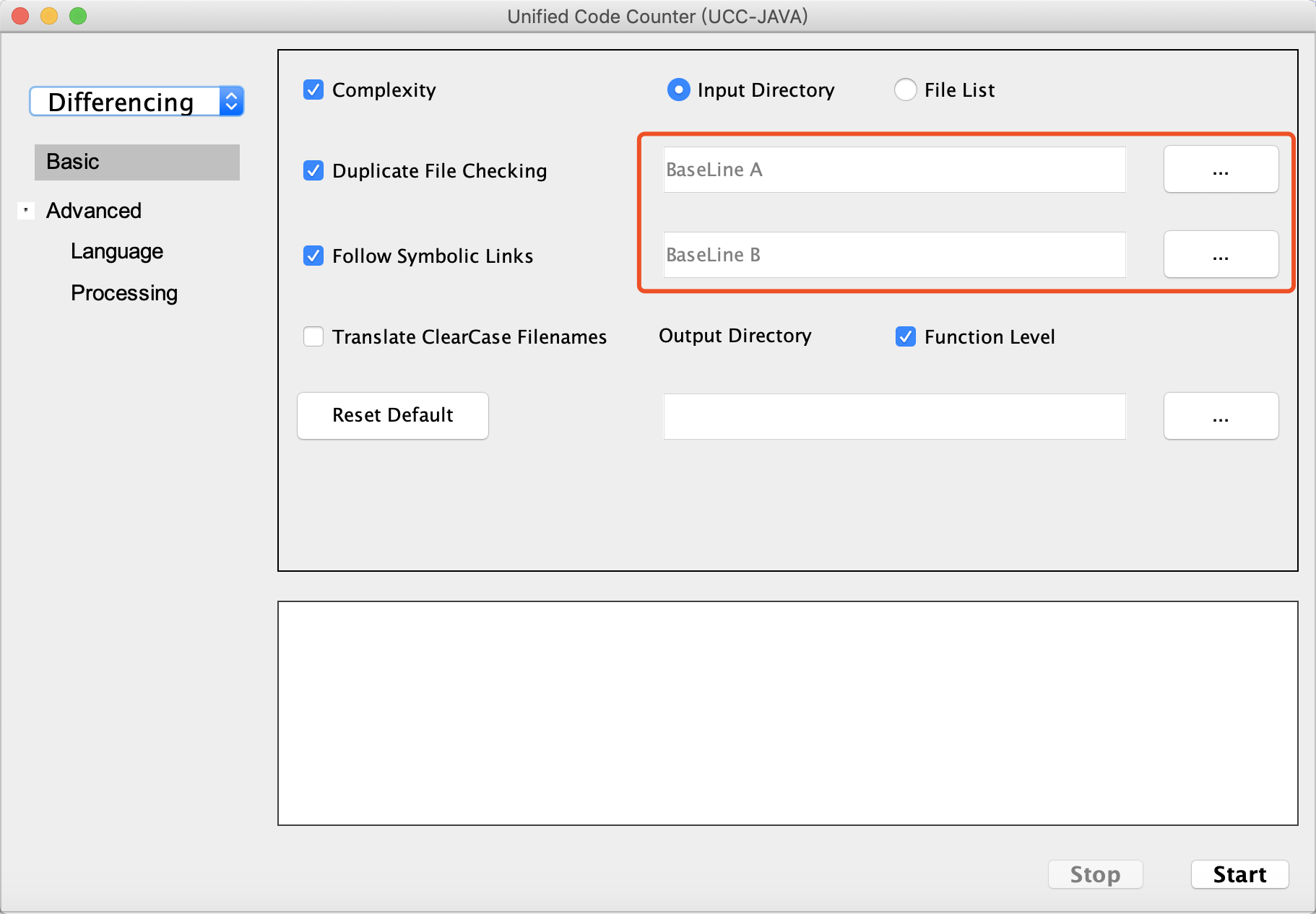


1. Customize the processing setting in advanced tab (optional)
2. Click “Start” to start processing, and click “Stop” if needed

**Differencing Example:**

1. **Set GUI function to “Differencing”**

****

1. Browse directory or file on local computer
2. The other part are the same with counting