

IccSorter Manual

Contents

Description.....	1
lccSorter	1
Installation.....	2
Logic File Description/Syntax.....	2
Logic File.....	2
Example Logic File	4
Definitions	4
Modifications.....	5

Description

This document describes how to use the lccSorter.

lccSorter

is a command line program to sort one to many columns in a file.

The file columns can be delimited by any character. You can also change the delimiter after the sorting has completed.

Text qualifiers can be read and produced, i.e. if a delimiter is within a column value, the program will expect it to be surrounded by double quotes and will surround it with double quotes when writing the sorted records.

You can also use the program to convert a file's delimiters with no sorting, just do not provide any lcc:sortCol keys.

Installation

- copy the lccSorter.exe to a folder
- create a Logic File
- run the lccSorter.exe lcc:logicPath [...logic file...]

Logic File Description/Syntax

A Logic File is a Tab delimited text file. Any lines not recognized as a valid Key/Value pair, will be ignore and can be used as remarks/other.

The Logic File uses the syntax.

Syntax: **[Key]** [tab] **[Value]** ... [tab] **[Value]**

Example: lcc:key value

Any extra tabs in a line after the expected ones are considered remarks and will be ignored. This is a nice way to document specific Key settings (see Log Levels in the Logic File example(s) for reference). Also, if you place a tab before a line, that will essentially make it a remark and will be ignored, which makes using/not using logic without removing quicker.

Any line not starting with an expected key is ignored, which makes placing remarks/formatting the logic easy.

Logic File

lcc:sourcePath (*mandatory, one per Logic File*)

What file will contain the source records.

Syntax: lcc:sourcePath [tab] [..path..]

Example: lcc:sourcePath \\ourserver\share\$\folder\file1.txt

lcc:sourceDelimiter (*optional, one per Logic File*)

What delimiter is used between the columns in the source file. If none is provided, a default TAB character will be used.

Any single character can be used.

Use "[tab]" for a tab character.

Syntax: `lcc:sourceDelimiter [tab] [...]`

Example: `lcc:sourceDelimiter ,`

Example #2: `lcc:sourceDelimiter [tab]`

lcc:targetPath (*mandatory, one per Logic File*)

What file will be written to with the sorted records. If you provide the same name as the lcc:sourcePath, that file will be replaced.

Syntax: `lcc:targetPath [tab] [..path..]`

Example: `lcc:targetPath \\ourserver\share$\folder\file2.txt`

lcc:targetDelimiter (*optional, one per Logic File*)

What delimiter is used between the columns in the target file. If none is provided, a default TAB character will be used.

Any single character can be used.

Use "[tab]" for a tab character.

Syntax: `lcc:targetDelimiter [tab] [...]`

Example: `lcc:targetDelimiter ,`

Example #2: `lcc:targetDelimiter [tab]`

lcc:sortCol (*mandatory, at least one per Logic File, but can be as many as you want*)

What column(s) to sort. Provide one of these keys for each column you want sorted.

Note: if sorting multiple columns, you should provide these keys in the order you want sorted. For example, if you want to sort names and have the final list be sorted by Last Name, then First Name, you would provide keys in the following order: First Name, Last Name. This is because you want to sort the primary column(s) last.

This key provide what column to sort, and in what order.

The order can be one of these values:

- ascending

- descending

Syntax: `lcc:sortCol [tab] [..column #..] [tab] [..order..]`

Example: `lcc:sortCol 1 ascending`

Example #2 (multiple):

```
lcc:sortCol 4 ascending
lcc:sortCol 2 descending
lcc:sortCol 1 ascending
lcc:sortCol 3 ascending
```

Example Logic File

This example Logic File will do the following:

- load records from file1.txt, looking for comma as a delimiter, into memory
- sort column #4 in ascending order
- sort column #2 in descending order
- sort column #1 in ascending order
- sort column #3 in ascending order
- write file2.txt with the sorted records, using a TAB as a delimiter

```
lcc:sourcePath file1.txt
```

```
lcc:sourceDelimiter ,
```

```
lcc:targetPath file2.txt
```

```
lcc:targetDelimiter [tab]
```

```
lcc:sortCol 4 ascending
```

```
lcc:sortCol 2 descending
```

```
lcc:sortCol 1 ascending
```

```
lcc:sortCol 3 ascending
```

Definitions

Modifications

NAME	DATE	MODIFICATION
David Mielcarek	20190711	Created

End of document