IccColumnsCruncher Manual

Contents

Description...........................................................................................................................................................................3

IccColumnsCruncher ............................................................................................................................................................3

Installation.............................................................................................................................................................................5

Using .......................................................................................................................................................................................5

Starting .....................................................................................................................................................................................5

Starting with an alternate Logic File......................................................................................................................................5

Logic File Description/Syntax....................................................................................................................................................5

Logic File – Category: Logging...............................................................................................................................................6

Logic File – Category: Debugging............................................................................................................................................9

Logic File – Category: Files....................................................................................................................................................9

Logic File – Category: Reports...............................................................................................................................................13

Logic File – Category: Actions...............................................................................................................................................15

Action Keys For: (Multi-Purpose) ......................................................................................................................................17

Action Keys For: Append.......................................................................................................................................................21

Action Keys For: Append Columns.......................................................................................................................................23

Action Keys For: Camel Case ...............................................................................................................................................24

Action Keys For: Capitalize ...................................................................................................................................................25

Action Keys For: Combine Duplicates, Append.....................................................................................................................26
Description
This document describes how to use the lccColumnsCruncher.
Lower Columbia College

is a Command Line program created to work with columns in one to many delimited source files (aka Data Sets).

The core purpose is to pull in one to many Data Sets, combine, manipulate the data, and create finalized exportable ‘master’ file(s). The power of its use comes from it performing each action sequentially. This allow you to do things like:

- sort on a column
- perform comparisons, translations, logic
- sort on another column
- do more actions
- etc.

This way, for example, you could sort on one column, have it bring in values from another Data Set per those values, then sort again on another column for another Data Set.

The program is controlled by a Logic File, from one to many different ones.

The program does the following:

- reads the Logic File
- loads Source Files into memory as Data Sets
- process actions against those Data Sets, building a Master Data Set (created in memory only)
- write the Master Data Set to Target File(s)
- produce reports
- produce stats

The current list of capabilities are (see the Logic File section for details on each capability):

- Load source files, on specified delimiter: loads file into memory for use in Actions
- File stats: shows stats on files loaded
- Combine sets: combine file records in memory together
- Record stats: show record stats
- Write targets: write final records to target files
- Processing Actions
- Log/write Sample Records: output sampling of records pre/post each action
- Action stats: report what actions were loaded and their settings

On Keys:

- lcc:actionColumnValue
- lcc:actionValue
special values can be replaced within the value when processing the Logic File. For example, you can supply the ‘[lcc:tab]’ in a value, to have a TAB character placed in that position. These special values will be replaced:

<table>
<thead>
<tr>
<th>Value</th>
<th>Replaced With</th>
</tr>
</thead>
<tbody>
<tr>
<td>[lcc:space]</td>
<td>Space Character</td>
</tr>
<tr>
<td>[lcc:tab]</td>
<td>TAB Character</td>
</tr>
</tbody>
</table>

**Installation**

- copy the lccColumnsCruncher.exe to a folder
- create a Logic File, depending on the need
- run the lccColumnsCruncher.exe

**Using**

Prerequisites: configure a Logic File.

The program will default to using a Logic File called:

```
lccColumnsCruncher-logic.txt
```

in the same folder as the program. You can change which Logic File to use by specifying a different one on the command line.

**Starting**

- run the lccColumnsCruncher.exe

**Starting with an alternate Logic File**

- run the `lccColumnsCruncher.exe lcc:logicPath [Logic File]`
- ex: `lccColumnsCruncher.exe lcc:logicPath ourLogic.txt`

**Logic File Description/Syntax**

Lower Columbia College
The 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]
Example: lcc:key value

Any extra tabs in a line after the first 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.

Though you can place the Key/Values in any order, the program will read/execute the settings/actions as they are loaded, per the category of key.

There are categories, as they are executed:
- Logging
- Debug
- Load Source Files
- Process Actions/Reports
- Write Target Files

To make creating/maintaining your logic file easier, we recommend placing your keys in the same order. i.e. place your Load File keys at top, following by Actions/Reports, then Target Files (please see Logic File example(s) for reference).

Logic File – Category: Logging

The Logging category instructs the program what to give feedback on. Most feedback is logged to a log file, but, some basic information is also copied to the screen.

The first setting in your Logic File should be to determine what your log file path/name will be.

**lcc:logPath** *(optional, but highly recommended)*(one per Logic File)
The path/root name of the Log File. The program will place the '.log' extension automatically and will also append a Year/Month date.

Lower Columbia College
Example:  lcc:logPath  lccColumnsCruncherLog

lcc:logLevel (optional) (none to many per Logic File)

Specifies what log information will be logged.  Provide a new line for each level desired.

Log Levels

1  Loading the Logic File: display loading the Logic File (available on the command line only)
2  Check Logic: displays checks performed on the Keys supplied by the Logic File.
3  Processing Logic: display information about logic being processed.
4  Load Logic Keys/values: display the Keys/values being loaded (available on the command line only).
5  Files Ids loaded: the File Ids loaded.
6  Files Settings: file settings loaded from the Logic File.
7  Load Source Files: display loading source files.
8  Load File: display loading a source file.
9  Skipping All Files: display information when skip loading any source files.
10 File Path Found: display file paths found.
11 File Stats: stats about files loaded.
12 Combine Sets: display information about combining Data Sets.
13 Record Stats: display stats about records.
14 Write Targets: display information when writing target files.
15 Write Target: display information when writing a target file.
16 Target Column Stats: display stats about target columns.
17 Processing Actions: display information about Actions being processed.
18 Sort Column, Start/Stop: display when sorting columns is started/stopped.
19 Action Stats: display stat about actions processed.
20 Sort Column, Compares: display comparing columns for sorting records.
21 Sort Column, Moves: display when moving records.
22 Sort, Pause On Move: will pause between each move (used for testing).
23 Split On Character: display information about splitting a value on a character.
24 Parse File On Delimiter: display when parsing a file on a delimiter.
25 Checking Unique Ids: display when checking for Unique Ids (Logic File).
26 Capitalize: display information when capitalizing a value.
27 Swap Columns: display information when swapping columns.
28 Convert To: display information when converting a value into another format.
29 Camel Case: display information when converting a value into camel case.
30 Find Match And Insert: display information when finding a match and inserting into another.
31 Append: display information when appending to a value.
Prepend: display information when prepending to a value.
Check To Skip Record: display information when checking to skip a record.
Find Match And Replace: display information when finding a match and replacing another.
Math: Multiply: display information when performing a multiplication.
Insert Blank Columns: display information when insert blank columns.
Check Reports: display information when checking reports.
Display Record: display information when displaying a record.
Log/Write Sample Record: display information when outputting sample records.
Initialize Reports: display information when initializing reports.
Finish Reports: display information when finishing reports.
Math: If <=, Replace: display information when performing math check.
Math: If >= and <=, Replace: display information when performing math check.
Math: If >=, Replace: display information when performing math check.
Keep Records With Column Value: display information when keeping records matching a value.
Skip Records With Column Value: display information when skipping records matching a value.
File Records Loaded: display file records loaded.
Find Duplicates: display information when finding duplicates.
Show Skip Records In Target Files: display information when skipping records for target files.
Combine Duplicates: display information when combining duplicates.
Pad Left/Right: display information when padding a value.
Empty Column When Value Not In Supplied Values: display information when emptying column.
Pre Set ToLowerCase And Double Values: display information when creating lowercase and double values for comparisons/sorting.
Records Compare (for sort): display information when doing a records compare.
Not Empty, Place Value: display information when checking if not empty and placing a value.
Empty, Place Value: display information when checking if empty and placing a value.
Keep Record With Column Value Containing: display information when keeping records that have a column value containing a value.
Skip Record With Column Value Containing: display information when skipping records that have a column value containing a value.
Keep Records With Empty Column: display information when keeping records that have an empty column.
Keep Records With Not Empty Column: display information when keeping records that have a not empty column.
Skip Records With Empty Column: display information when skipping records that have an empty column.
Skip Records With Not Empty Column: display information when skipping records that have a not empty column
Upper Case: display information when converting a value into upper case
Lower Case: display information when converting a value into lower case
Append Columns: display information when appending one column to another
Remove Columns: display information when removing columns
Trim: display information when removing white-space characters from left and right
Remove Consecutive Spaces: display information when removing multiple spaces when consecutive
Replace: display information when replacing values inside a column value
Insert Line At Each Record Set: display information when inserting a new line (record) after each new record set
Split On Position: display information about splitting a value on a position.

Syntax: lcc:logLevel [tab] [#]

Example: lcc:logLevel 5
Example: lcc:logLevel 5
Example: lcc:logLevel 12

Logic File – Category: Debugging

The Debugging category instructs the program to do global changes for running in a debug mode. These are usually only done by the programmer making this program, but, available if needed.

lcc:debugLevel (optional) (none to many per Logic File)
Specifies what debugging changes will take effect. Provide a new line for each level desired.

Debug Levels
1 Skip All Files: skip all files and display information when skipping all files.

Logic File – Category: Files

The Files category instructs the program what files are being used.
There are two types of files:
- **Source**: a file that is going to be loaded into memory as a Data Set.
- **Target**: a file that is going to be created/used for output.

Each File is defined by creating a File Settings Set (FSS).

Each FSS is a File defined using Keys starting with: `lcc:file`

Each FSS starts with the Key: `lcc:fileId`

- **lcc:fileId** *(mandatory) (one per FSS)*
  The ID used for accessing this File with Actions. Each File Id must be unique, but, can be any value. Should describe what the file contains.
  
  Syntax: `lcc:fileId [tab] [value]`
  
  Example: `lcc:fileId State Registered Students`

- **lcc:fileType** *(mandatory) (one per FSS)*
  The type of File.
  
  Valid values
  - **Source**
  - **Target**
  
  Syntax: `lcc:fileType [tab] [Type]`
  
  Example: `lcc:fileType Source`

- **lcc:filePath** *(mandatory) (one per FSS)*
  The path of file.
  
  Syntax: `lcc:filePath [tab] [Path]`
  
  Example: `lcc:filePath E:\folder\ourFile.txt`
  
  Example: `lcc:filePath \server\share$\ourFile.txt`

- **lcc:fileDelimiter** *(mandatory) (one per FSS)*
  The delimiter character used between the columns. If more than one characters supplied, only the first is used, i.e. if 'dog' is supplied, 'd' would be the delimiter.

Lower Columbia College
You can also specify these white-space characters:

- \[\text{tab}\]: will change to a single tab character with processing
- \[\text{space}\]: will change to a single space character with processing

**Requirement:** must be a single character, or a White Space key word.

**Syntax:** `lcc:fileDelimiter \[\text{tab}\] [Delimiter]`

**Example:**
- `lcc:fileDelimiter`
- `lcc:fileDelimiter \[\text{tab}\]`

**lcc:fileRecordsMaxLoad (optional) (one per FSS)**

Control how many records are loaded. This is beneficial if you have a large file and only want to load a few records first for testing your Logic.

**Syntax:** `lcc:fileRecordsMaxLoad \[\text{tab}\] [\#]`

**Example:**
- `lcc:fileRecordsMaxLoad 50`

**lcc:fileRecordsSkipLoading (optional) (one per FSS)**

Skip first records from being processed (like headers, or if you want to jump over a set of records).

**Syntax:** `lcc:fileRecordsSkipLoading \[\text{tab}\] [\#]`

**Example:**
- `lcc:fileRecordsSkipLoading 1`

**lcc:fileTrimColumns (optional) (one per FSS)**

Remove whitespace characters from the left and right of each value.

**Valid values**

- YES

**Syntax:** `lcc:fileTrimColumns \[\text{tab}\] [YES]`

**Example:**
- `lcc:fileTrimColumns YES`

**lcc:fileLeftTrimColumns (optional) (one per FSS)**

Remove whitespace characters from the left of each value.

**Valid values**

Lower Columbia College
lcc:fileLeftTrimColumns (optional)(one per FSS)
Remove whitespace characters from the right of each value.

Valid values
- YES

Syntax: lcc:fileLeftTrimColumns [tab] [YES]
Example: lcc:fileLeftTrimColumns YES

lcc:fileRightTrimColumns (optional)(one per FSS)
Remove whitespace characters from the right of each value.

Valid values
- YES

Syntax: lcc:fileRightTrimColumns [tab] [YES]
Example: lcc:fileRightTrimColumns YES

lcc:fileShowRecords (optional)(one per FSS)
Show records as they are loaded from the file.

Valid values
- YES

Syntax: lcc:fileShowRecords [tab] [YES]
Example: lcc:fileShowRecords YES

lcc:fileAddHeaderRecord (optional)(one to many per FSS)
Valid for File Types: Target
Add columns headers. Add this key for each column header you want added.

Syntax: lcc:fileAddHeaderRecord [tab] [...] 
Example: lcc:fileAddHeaderRecord First Name
lcc:fileAddHeaderRecord Middle Name/Initial
lcc:fileAddHeaderRecord Last Name

lcc:fileSkip (optional)(one per FSS)
Whether to use or skip this File. This use beneficial if you want to keep the logic settings in place for a file, but, do not use it.
Valid values

- YES

Syntax:  lcc:fileSkip [tab] [YES]
Example:  lcc:fileSkip  YES

Example File Set

lcc:fileId    HPSA Students
lcc:fileType  Source
lcc:filePath  hpsaStudents.txt
lcc:fileDelimiter  [tab]
lcc:fileRecordsMaxLoad  10
lcc:fileTrimColumns  YES
lcc:fileSkip  YES

Logic File – Category: Reports

The Files category instructs the program what reports should be created.

There are two types of reports:
- Show Pre/Post Action Record Samples: output sample records before/after each action.
- Show Completed Records: output completed records after all actions finished.

Each Report is defined by creating a Report Settings Set (RSS).

Each RSS Report is defined using Keys starting with: lcc:report

Each RSS starts with the Key: lcc:reportId

lcc:reportId  (mandatory)(one per FSS)

The ID used for creating this Report. Each Report Id must be unique, but, can be any value. Should describe what the Report contains.

Lower Columbia College
Syntax: \texttt{lcc:reportId [tab] [value]}
Example: \texttt{lcc:reportId} Show record as it changes

\textbf{lcc:reportType} \textit{(mandatory)(one per FSS)}

The type of Report.

**Valid values** (types of reports)
- Show Pre/Post Action Record Samples
- Show Completed Records

Syntax: \texttt{lcc:reportType [tab] [Type]}
Example: \texttt{lcc:reportType} Show Pre/Post Action Record Samples

\textbf{lcc:reportSkip} \textit{(optional)(one per FSS)}

Whether to use or skip this Report. This use beneficial if you want to keep the logic settings in place for a file, but, do not use it.

**Valid values**
- YES

Syntax: \texttt{lcc:reportSkip [tab] [YES]}
Example: \texttt{lcc:reportSkip} YES

Example Report Set: Sample Record
\texttt{lcc:reportId} Show record as it changes
\texttt{lcc:reportType} Show Pre/Post Action Record Samples
\texttt{lcc:reportFile} sampleRecords.txt
\texttt{lcc:reportRecordNumber} 1
\texttt{lcc:reportRecords} 10
\texttt{lcc:reportSkip} YES

Example Report Set: All finished records, including skipped
\texttt{lcc:reportId} Show all records after program finished
\texttt{lcc:reportType} Show Completed Records
Logic File – Category: Actions

The Actions category instructs the program what actions should be performed on Records.

Valid Values (types of actions)

- **Append**: append value to column value.
- **Append Columns**: append the value of one column to another.
- **Camel Case**: camel case the column value.
- **Capitalize**: capitalize the column value.
- **Combine Duplicates, Append**: combine duplicate records and append non-unique values.
- **Combine Duplicates, Keep First**: combine duplicate records and keep values from first record.
- **Combine Duplicates, Keep Last**: combine duplicate records and keep values from last record.
- **Combine Sets**: combine records from one Data Set to end of another Data Set.
- **Convert To**: convert column value into another data format.
- **Empty A Column When Another Column Value Not In Supplied Values**: delete a column value.
- **Find Duplicates**: find duplicate records.
- **Find Match And Insert**: find record column value from another Data Set and insert column value.
- **Find Match And Replace**: find record column value from another Data Set and replace column value.
- **If Empty, Place Value**: if a column value is empty, place a value.
- **If Not Empty, Place Value**: if a column value is not empty, place a value.
- **Insert Blank Columns**: insert blanks columns.
- **Insert Line At Each Record Set**: insert a new line (record) after each new record set.
- **Keep Records With Column Value**: keep records that match certain value(s).
- **Keep Records With Column Value Containing**: keep records that contain certain value(s).
- **Keep Records With Empty Column**: keep records that have an empty column.
- **Keep Records With Not Empty Column**: keep records that have a not empty column.
- **Lower Case**: lower case the column value.
- **Math: if <=, replace**: if column value is <= a number value, replace a column value.
- **Math: if >= and <=, replace**: if column value is >= and <= a number value, replace a column value.
- **Math: if >=, replace**: if column value is >= a number value, replace a column value.
- **Math: Multiply**: multiple the column value by a number.
- **Pad Left**: left pad a column value with a character.
- **Pad Right**: right pad a column value with a character.
- **Prepend**: prepend a column value with a value.
- **Remove Columns**: remove columns.
- **Remove Consecutive Spaces**: remove spaces where there are more than one in a consecutive position.
- **Replace**: replace a value inside a column with another value.
- **Skip Records With Column Value**: skip records that match certain value(s).
- **Skip Records With Column Value Containing**: keep records that contain certain value(s).
- **Skip Records With Empty Column**: skip records that have an empty column.
- **Skip Records With Not Empty Column**: skip records that have a not empty column.
- **Sort**: sort on a column.
- **Split On Character**: split a column value on a character.
- **Split On Position**: split a column value on a position.
- **Swap Columns**: swap columns.
- **Trim**: remove white-space characters from left and right of the column value.
- **Trim Left**: remove white-space characters from the left of the column value.
- **Trim Right**: remove white-space characters from the right of the column value.
- **Upper Case**: upper case the column value.

Each Action is defined by creating an Action Settings Set (ASS).

Each ASS Action is defined using Keys starting with: `lcc:action`

Each ASS starts with the Key: `lcc:actionId`

**lcc:actionId** *(mandatory)(one per ASS)*

The ID used for processing this Action. Each Action Id must be unique, but, can be any value. Should describe what the Action will do.

**Syntax:** `lcc:actionId [tab] [value]`

**Example:** `lcc:actionId  Add File to master`
**Action Keys For: (Multi-Purpose)**

**lcc:actionSetId (mandatory)(one per ASS)**

The ID to specify which Data Set is being used. Each Action Set Id must reference a loaded File or the 'Master'. Master is a unique ID that cannot be assigned, but, the records are available to add to, manipulate. The 'Master' set is empty by default, you must use the 'Combine Set' action to add records to it.

**Syntax:**

```
lcc:actionSetId [tab] [value]
```

**Example:**

```
lcc:actionSetId State Registered Students
```

```
lcc:actionSetId Master
```

**lcc:actionSet2Id (mandatory)(one per ASS)**

The ID to specify which 2nd Data Set is being used. A 2nd Data Set is used when pulling from, sending to, or comparing records between Data Sets. Each Action Set Id must reference a loaded File or the 'Master'. Master is a unique ID that cannot be assigned, but, the records are available to add to, manipulate. The 'Master' set is empty by default, you must use the 'Combine Set' action to add records to it.

**Syntax:**

```
lcc:actionSet2Id [tab] [value]
```

**Example:**

```
lcc:actionSet2Id State Registered Students
```

```
lcc:actionSet2Id Master
```

**lcc:actionFlag (optional)(none to many per ASS)**

Flags are used to further control an action. Provide one key/value for each flag you want to implement on an Action.

**Valid values**

- **Include Skipped Records**
  - **Actions Can Be Used On:**
    - **All**
      - Keep records marked for skip, including ones marked for 'skip' by previous actions.

- **Do Not Mark Skip**
  - **Actions Can Be Used On:**
    - **All**
      - When performing 'Keep Records ...' Action, will not mark records unmatched as Skip.

- **Do Not Mark Keep**
  - **Actions Can Be Used On:**
    - **All**
      - When performing 'Skip Records ...' Action, will not mark records unmatched as Keep.
- **DecimalPlaces:#**
  - **Actions Can Be Used On:**
    - Math: Multiply
    Truncates and/or forces a set # of decimal places. The only valid numbers of decimal places are from zero (0) and then any positive number.
    - This value has two parts, each part is separated by a colon ':'. The first part is always 'DecimalPlaces'.
    - For example, if you want 5 decimals, use DecimalPlaces:5

- **Right To Left**
  - **Actions Can Be Used On:**
    - Split On Character
    When splitting value, splits will start search on the right side of the value. This also affects the Min/Max (columns) flags. For example, if you split:
      - David Programmer Mielcarek
    on space normally, the split columns would be:
    1. David
    2. Programmer
    3. Mielcarek
    And if you provided a 'Max' columns key with 2, then you would only get:
    1. David
    2. Programmer
    However, if you supply this Right To Left flag, with the same 'Max' key, you would get:
    1. Mielcarek
    2. Programmer

- **Keep Left Over**
  - **Actions Can Be Used On:**
    - Split On Character
    When splitting value, if a max is defined, this will copy the left over portion of the value into a new column to the right. If there is nothing left over, a blank column will be inserted. For example, if you split on space for:
      - David walked along the beach
    on a space character and set a maximum split of 2, the split columns would be:
    1. David
    2. Walked
    and if this key was provided, then a 3rd column would be inserted with the left over value, resulting in:
    1. David
    2. Walked
3. along the beach

**Syntax:**  \[\text{lcc:actionFlag} \{tab\} \{value\}\]

**Example:**  \[\text{lcc:actionFlag} \text{Include Skipped Records}\]

**Example:**  \[\text{lcc:actionFlag} \text{Include Skipped Records}\]

**Example:**  \[\text{lcc:actionFlag} \text{Do Not Mark Skip}\]

**Example:**  \[\text{lcc:actionFlag} \text{DecimalPlaces:5}\]

\[\text{lcc:actionType} \ (mandatory)\ (one per ASS)\]

The type of Action.

**Valid values**
- Append
- Append Columns
- Camel Case
- Capitalize
- Combine Duplicates, Append
- Combine Duplicates, Keep First
- Combine Duplicates, Keep Last
- Combine Sets
- Convert To
- Empty A Column When Another Column Value Not In Supplied Values
- Find Duplicates
- Find Match And Insert
- Find Match And Replace
- If Empty, Place Value
- If Not Empty, Place Value
- Insert Blank Columns
- Keep Records With Column Value
- Keep Records With Column Value Containing
- Keep Records With Empty Column
- Keep Records With Not Empty Column
- Lower Case
- Math: if <=, replace
- Math: if >= and <=, replace
- Math: if >=, replace
- Math: Multiply
- Pad Left
- Pad Right
- Prepend
- Remove Columns
- Remove Consecutive Spaces
- Skip Records With Column Value
- Skip Records With Column Value Containing
- Skip records with Empty Column
- Skip records with Not Empty Column
- Sort
- Split On Character
- Split On Position
- Swap Columns
- Trim
- Trim Left
- Trim Right
- Upper Case

Syntax: lcc:actionType [tab] [Type]
Example: lcc:actionType Combine Sets

lcc:actionColumnValueMinLenToApplyRule (optional)(one per ASS)
If set, will only effect column value if the value has this minimum length.
Syntax: lcc:actionColumnValueMinLenToApplyRule [tab] [#]
Example: lcc:actionColumnValueMinLenToApplyRule 1

lcc:actionColumnValueMaxLenToApplyRule (optional)(one per ASS)
If set, will only effect column value if the value has this maximum length.
Syntax: lcc:actionColumnValueMaxLenToApplyRule [tab] [#]
Example: lcc:actionColumnValueMaxLenToApplyRule 1

Lower Columbia College
**lcc:actionShowSkippedRecords** *(optional)(one per ASS)*
If an action would skip records, show the records skipped.

**Valid values**
- YES

**Syntax:**
lcc:actionShowSkippedRecords [tab] [YES]

**Example:**
lcc:actionShowSkippedRecords YES

**lcc:actionSortMatchAsNumber** *(optional)(one per ASS)*
If set, will treat the Columns being affected by the Action as a number. For example, if not set the value '9123' would sort after and not match '009123'. If set, both '9123' and '009123' would be considered the same.

**Valid values**
- YES

**Syntax:**
lcc:actionSortMatchAsNumber [tab] [YES]

**Example:**
lcc:actionSortMatchAsNumber YES

**lcc:actionSkip** *(optional)(one per FSS)*
Whether to use or skip this Action. This use beneficial if you want to keep the logic settings in place for a file, but, do not use it.

**Valid values**
- YES

**Syntax:**
lcc:actionSkip [tab] [YES]

**Example:**
lcc:actionSkip YES

**Action Keys For: Append**

**Additional Keys that can be used from the Multi-Purpose Keys:**
- lcc:actionId  *mandatory*
- lcc:actionType  *mandatory*
- lcc:actionSetId  mandatory
- lcc:actionColumnValueMinLenToApplyRule  optional
- lcc:actionColumnValueMaxLenToApplyRule  optional
- lcc:actionFlag  optional
- lcc:actionSkip  optional

**lcc:actionColumnId** (mandatory) (one per ASS)

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

**Syntax:** lcc:actionColumnId [tab] [ID]

**Example:** lcc:actionColumnId Email

**lcc:actionColumnPos** (mandatory) (one per ASS)

The column position number, where '1' is the first column, '2' is second, etc. This value will be appended to.

**Requirement:** must be a number

**Syntax:** lcc:actionColumnPos [tab] [#]

**Example:** lcc:actionColumnPos 11

**lcc:actionColumnValue** (mandatory) (one per ASS)

What value should be appended.

**Syntax:** lcc:actionColumnValue [tab] [...]

**Example:** lcc:actionColumnValue @my.lowercolumbia.edu

**Example Action Set: Append**

lcc:actionId  Append email domain @my.lowercolumbia.edu
lcc:actionType  Append
lcc:actionSetId  Master
lcc:actionColumnId  Email
lcc:actionColumnPos 11
lcc:actionColumnValue  @my.lowercolumbia.edu
lcc:actionColumnValueMinLenToApplyRule  1
lcc:actionSkip  YES
Action Keys For: Append Columns

Additional Keys that can be used from the Multi-Purpose Keys:

- `lcc:actionId` mandatory
- `lcc:actionType` mandatory
- `lcc:actionSetId` mandatory
- `lcc:actionColumnValueMinLenToApplyRule` optional
- `lcc:actionColumnValueMaxLenToApplyRule` optional
- `lcc:actionFlag` optional
- `lcc:actionSkip` optional

**lcc:actionColumnId** (mandatory)(one per ASS)
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax: `lcc:actionColumnId [tab] [ID]`
Example: `lcc:actionColumnId Last Name`

**lcc:actionColumnPos** (mandatory)(one per ASS)
The column position number, where '1' is the first column, '2' is second, etc. What column value will be appended to.

Requirement: must be a number

Syntax: `lcc:actionColumnPos [tab] [#]`
Example: `lcc:actionColumnPos 1`

**lcc:actionColumn2Id** (mandatory)(one per ASS)
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

Syntax: `lcc:actionColumn2Id [tab] [ID]`
Example: `lcc:actionColumn2Id First Name`

**lcc:actionColumn2Pos** (mandatory)(one per ASS)
The column position number, where '1' is the first column, '2' is second, etc. What column value will be appended from.

Requirement: must be a number

Syntax: `lcc:actionColumn2Pos [tab] [#]`
Example: `lcc:actionColumn2Pos 2`
**lcc:actionColumnValue** *(mandatory)* *(one per ASS)*

What will be a separator between the two column values. If both columns have a value, then the separator will be place between them on appending. For example if you append to "David" a column with "Mielcarek" and supply this value with "--><", the first column will end up with a value of "David--><Mielcarek". If you want just a space, only put a space in the value (not the code "[space]", but, an actual space).

*Requirement: must be a number*

*Syntax: lcc:actionColumnValue [tab] [...]*

*Example: lcc:actionColumnValue --:--*

---

**Example Action Set: Append Columns**

```
  lcc:actionId      Append Program Description and Program Title
  lcc:actionType    Append Columns
  lcc:actionSetId   Master
  lcc:actionColumnId Program Description
  lcc:actionColumnPos 22
  lcc:actionColumn2Id Program Title
  lcc:actionColumn2Pos 23
  lcc:actionColumnValue --:--
  lcc:actionSkip    YES
```

---

**Action Keys For: Camel Case**

Additional Keys that can be used from the Multi-Purpose Keys:

- **lcc:actionId** mandatory
- **lcc:actionType** mandatory
- **lcc:actionSetId** mandatory
- **lcc:actionColumnValueMinLenToApplyRule** optional
- **lcc:actionColumnValueMaxLenToApplyRule** optional
- **lcc:actionFlag** optional
- **lcc:actionSkip** optional

**lcc:actionColumnId** *(mandatory)* *(one per ASS)*
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax: `lcc:actionColumnId [tab] [ID]`
Example: `lcc:actionColumnId Street`

**lcc:actionColumnPos (mandatory)(one per ASS)**
The column position number, where '1' is the first column, '2' is second, etc. This value will be camel cased.

*Requirement: must be a number*

Syntax: `lcc:actionColumnPos [tab] [#]`
Example: `lcc:actionColumnPos 7`

Example Action Set: Camel Case

```
lcc:actionId Camel Case - Street
lcc:actionType Camel Case
lcc:actionSetId Master
lcc:actionColumnId Street
lcc:actionColumnPos 7
lcc:actionSkip YES
```

**Action Keys For: Capitalize**

Additional Keys that can be used from the Multi-Purpose Keys:

- `lcc:actionId` mandatory
- `lcc:actionType` mandatory
- `lcc:actionSetId` mandatory
- `lcc:actionColumnValueMinLenToApplyRule` optional
- `lcc:actionColumnValueMaxLenToApplyRule` optional
- `lcc:actionFlag` optional
- `lcc:actionSkip` optional

**lcc:actionColumnId (mandatory)(one per ASS)**
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax: `lcc:actionColumnId [tab] [ID]`
Example: lcc:actionColumnId First Name

**lcc:actionColumnPos** *(mandatory) (one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. This value will be capitalized.

*Requirement*: must be a number

*Syntax*: lcc:actionColumnPos [tab] [#]

*Example*: lcc:actionColumnPos 1

Example Action Set: Capitalize

<table>
<thead>
<tr>
<th>lcc:actionId</th>
<th>Capitalize - First Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>lcc:actionType</td>
<td>Capitalize</td>
</tr>
<tr>
<td>lcc:actionSetId</td>
<td>Master</td>
</tr>
<tr>
<td>lcc:actionColumnId</td>
<td>First Name</td>
</tr>
<tr>
<td>lcc:actionColumnPos</td>
<td>1</td>
</tr>
<tr>
<td>lcc:actionSkip</td>
<td>YES</td>
</tr>
</tbody>
</table>

**Action Keys For: Combine Duplicates, Append**

Additional Keys that can be used from the Multi-Purpose Keys:

- lcc:actionId *(mandatory)*
- lcc:actionType *(mandatory)*
- lcc:actionSetId *(mandatory)*
- lcc:actionColumnValueMinLenToApplyRule *(optional)*
- lcc:actionColumnValueMaxLenToApplyRule *(optional)*
- lcc:actionShowSkippedRecords *(optional)*
- lcc:actionSortMatchAsNumber *(optional)*
- lcc:actionFlag *(optional)*
- lcc:actionSkip *(optional)*

**lcc:actionColumnValue** *(mandatory) (one to many per ASS)*

What column(s) should be used to determine duplicates. These columns must be sorted to correctly find duplicates. You can use the Action 'Sort' just prior to this action to sort the columns.
If you wish to use more than one column to determine duplicates, make sure to sort all of those columns and specify one of these keys for each column.

*Requirement: must be a number*

Syntax: `lcc:actionColumnValue [tab] [#]`
Example: `lcc:actionColumnValue 4`

**lcc:actionColumnDelimiter** *(mandatory) (one per ASS)*

What character to use to separate column values when appended. When duplicates are found, the following logic applies:

- check each column between the two records
- if they match, only keep one of those values
- if they do not match, and both contain values, append together, separated by the delimiter
- if they do not match, and only one contains a value, do not use the delimiter

*Requirement: must be a single character*, or a White Space key word.

Syntax: `lcc:actionColumnDelimiter [tab] [Character]`
Example: `lcc:actionColumnDelimiter :`

Example Action Set: Combine Duplicates, Append

```plaintext
lcc:actionId Combine Duplicates - Previous Degree(s)
lcc:actionType Combine Duplicates, Append
lcc:actionSetId Master
lcc:actionColumnValue 4
lcc:actionDelimiter |
    lcc:actionShowSkippedRecords YES
    lcc:actionSkip YES
```

**Action Keys For: Combine Duplicates, Keep First**

Additional Keys that can be used from the Multi-Purpose Keys:

- `lcc:actionId` *mandatory*
- `lcc:actionType` *mandatory*
- `lcc:actionSetId` *mandatory*
- `lcc:actionColumnValueMinLenToApplyRule` *optional*
**lcc:actionColumnValue (mandatory) (one to many per ASS)**

What column(s) should be used to determine duplicates. These columns must be sorted to correctly find duplicates. You can use the Action 'Sort' just prior to this action to sort the columns.

If you wish to use more than one column to determine duplicates, make sure to sort all of those columns and specify one of these keys for each column.

*Requirement: must be a number*

**Syntax:**  
`lcc:actionColumnValue [tab] [#]`

**Example:**  
`lcc:actionColumnValue 4`

---

**Example Action Set: Combine Duplicates, Keep First**

<table>
<thead>
<tr>
<th>Action</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>lcc:actionId</td>
<td>Combine Duplicates - Previous Degree(s)</td>
</tr>
<tr>
<td>lcc:actionType</td>
<td>Combine Duplicates, Keep First</td>
</tr>
<tr>
<td>lcc:actionSetId</td>
<td>Master</td>
</tr>
<tr>
<td>lcc:actionColumnValue</td>
<td>4</td>
</tr>
<tr>
<td>lcc:actionShowSkippedRecords</td>
<td>YES</td>
</tr>
<tr>
<td>lcc:actionSkip</td>
<td>YES</td>
</tr>
</tbody>
</table>

---

**Action Keys For: Combine Duplicates, Keep Last**

Additional Keys that can be used from the Multi-Purpose Keys:

- `lcc:actionId` *mandatory*
- `lcc:actionType` *mandatory*
- `lcc:actionSetId` *mandatory*
- `lcc:actionColumnValueMinLenToApplyRule` *optional*
- `lcc:actionColumnValueMaxLenToApplyRule` *optional*
- `lcc:actionShowSkippedRecords` *optional*
lcc:actionColumnValue *(mandatory) (one to many per ASS)*

What column(s) should be used to determine duplicates. These columns must be sorted to correctly find duplicates. You can use the Action 'Sort' just prior to this action to sort the columns.

If you wish to use more than one column to determine duplicates, make sure to sort all of those columns and specify one of these keys for each column.

*Requirement: must be a number*

*Syntax: lcc:actionColumnValue [tab] [#]*

*Example: lcc:actionColumnValue 4*

**Example Action Set: Combine Duplicates, Keep Last**

**lcc:actionId** Combine Duplicates - Previous Degree(s)

**lcc:actionType** Combine Duplicates, Keep Last

**lcc:actionSetId** Master

**lcc:actionColumnValue** 4

**lcc:actionShowSkippedRecords** YES

**lcc:actionSkip** YES

**Action Keys For: Combine Sets**

**Additional Keys that can be used from the Multi-Purpose Keys:**

- **lcc:actionId** *mandatory*
- **lcc:actionType** *mandatory*
- **lcc:actionSetId** *mandatory*
- **lcc:actionSet2Id** *mandatory*
- **lcc:actionFlag** *optional*
- **lcc:actionSkip** *optional*

This action combines Set1Id (Data Set) to the end of Set2Id (Data Set #2).
Example Action Set: Combine Sets

lcc:actionId    Add File to master
lcc:actionType   Combine Sets
lcc:actionSetId  HPSA Students
lcc:actionSet2Id Master
    lcc:actionSkip YES

Action Keys For: Convert To

Additional Keys that can be used from the Multi-Purpose Keys:

- lcc:actionId    mandatory
- lcc:actionType   mandatory
- lcc:actionSetId  mandatory
- lcc:actionColumnValueMinLenToApplyRule  optional
- lcc:actionColumnValueMaxLenToApplyRule   optional
- lcc:actionFlag    optional
- lcc:actionSkip    optional

lcc:actionColumnId (mandatory) (one per ASS)
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.
Syntax: lcc:actionColumnId [tab] [ID]
Example: lcc:actionColumnId  Convert DOB To MM-DD-YYYY

lcc:actionColumnPos (mandatory) (one per ASS)
The column position number, where '1' is the first column, '2' is second, etc. This value will be converted to a different data format.
Requirement: must be a number
Syntax: lcc:actionColumnPos [tab] [#]
Example: lcc:actionColumnPos 1

lcc:actionColumnOriginalFormat (mandatory) (one per ASS)

Lower Columbia College
What format the column value is currently in. If the value does not match the format, and that format is required to convert, it will not be converted. Some conversions do not require an original format, like removing all characters except number.

**Valid values**
- **YY**: ex - 16
- **YYMMDD**: ex - 160323
- **YYYYMMDD**: ex - 20160323
- **MM/DD/YYYY**: ex - 03/23/2016
- **HHMMA/PM**: ex - 0320P
- **Microsoft Time Stamp**: ex - 131019629366366931
- **Unix Time Stamp**: ex - 1458730894
- *** (used when the original format does not matter)**

**Syntax:** \texttt{lcc:actionColumnOriginalFormat [tab] [FORMAT]}

**Example:** \texttt{lcc:actionColumnOriginalFormat MM/DD/YYYY}

**lcc:actionColumnNewFormat** *(mandatory)* *(one per ASS)*

What format the column value should be converted to.

**Valid values**
- **HH:MM AM/PM**
- **MM-DD-YYYY**
- **MM-DD-YY**
- **MM/DD/YYYY**
- **MM/DD/YYYY HH:MM:SS**
- **MM/DD/YY**
- **YYYYMMDD**
- **YYYY-MM-DD**
- **YYMMDD**
- **Numbers Only** *(removes all characters that are not numbers)*
- **Numbers And A Single Period Only** *(removes all characters that are not numbers, allowing a single period, i.e. decimal number)*
- **Phone Number ###-###-####**
- **Phone Number (###) ###-####**
- Phone Number (###) ####-####
- Phone Number (###) ############
- Phone Number ####.####.####
- Phone Number ###.###.####

Syntax:  `lcc:actionColumnNewFormat [tab] [FORMAT]
**Example:**  `lcc:actionColumnNewFormat YYYYMMDD`  
**Example:**  `lcc:actionColumnNewFormat Numbers Only`  
**Example:**  `lcc:actionColumnNewFormat Numbers And A Single Period Only`  

**lcc:actionTwoToFourYearThreshold** *(optional) (one per ASS)*

The threshold where a two to four year conversion assumes it should be this year, else it will use the previous year. If the value is at or below the threshold, it will apply. For example, if a two digit year is in the value as '14', and you have a threshold of '15', with a New Format of YYYY, the final value will show '2015'. With a threshold of '16', the same action would make a final value of '1915'.

You can also supply 'Current' to have it auto fill-in the current year, i.e. if today's date was 6/16/2016, it would auto place '16' in this value.

**Requirement:** must be a number or 'Current'

Syntax:  `lcc:actionTwoToFourYearThreshold [tab] [#]
**Example:**  `lcc:actionTwoToFourYearThreshold 15`  
**Example #2:**  `lcc:actionTwoToFourYearThreshold Current`  

Example Action Set: Convert To

`lcc:actionId  Convert DOB To MM-DD-YYYY`  
`lcc:actionType  Convert To`  
`lcc:actionSetId  Master`  
`lcc:actionColumnId  DOB`  
`lcc:actionColumnPos 5`  
`lcc:actionColumnOriginalFormat  YYMMDD`  
`lcc:actionColumnNewFormat  MM-DD-YYYY`  
`lcc:actionTwoToFourYearThreshold  15`  
`lcc:actionSkip YES`
Action Keys For: Empty Column When Another Column Value Not Supplied

Additional Keys that can be used from the Multi-Purpose Keys:

- **lcc:actionId**  *mandatory*
- **lcc:actionType**  *mandatory*
- **lcc:actionSetId**  *mandatory*
- **lcc:actionColumnValueMinLenToApplyRule**  *optional*
- **lcc:actionColumnValueMaxLenToApplyRule**  *optional*
- **lcc:actionFlag**  *optional*
- **lcc:actionSkip**  *optional*

**lcc:actionColumnId**  *(mandatory)(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

**Syntax:**  
lcc:actionColumnId [tab] [ID]

**Example:**  
lcc:actionColumnId High School Graduate

**lcc:actionColumnPos**  *(mandatory)(one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. This column will be used to see if a supplied value exists.

*Requirement: must be a number*

**Syntax:**  
lcc:actionColumnPos [tab] [#]

**Example:**  
lcc:actionColumnPos 21

**lcc:actionColumn2Id**  *(mandatory)(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

**Syntax:**  
lcc:actionColumn2Id [tab] [ID]

**Example:**  
lcc:actionColumn2Id High School Code

**lcc:actionColumn2Pos**  *(mandatory)(one per ASS)*

Lower Columbia College
The column position number, where '1' is the first column, '2' is second, etc. This column will be emptied if the supplied value exists.

**Requirement:** must be a number

**Syntax:** `lcc:actionColumn Pos [tab] [#]`

**Example:** `lcc:actionColumnPos 21`

**lcc:actionColumnValue** *(mandatory)* *(one to many per ASS)*

What value should be supplied. If more than one, supply this key for each value.

**Syntax:** `lcc:actionColumnValue [tab] [...]

**Example:** `lcc:actionColumnValue Y`

**Example:** `lcc:actionColumnValue N`

---

**Example Action Set:** Empty Column When Another Column Value Not Supplied

```
lcc:actionId   Empty High School Code value for High School Grad != Y
lcc:actionType Empty Column When Another Column Value Not Supplied
lcc:actionSetId Master
lcc:actionColumnId High School Graduate
lcc:actionColumnPos 21
lcc:actionColumn2Id High School Code
lcc:actionColumn2Pos 19
lcc:actionColumnValue Y
lcc:actionSkip YES
```

---

**Action Keys For: Find Duplicates**

**Additional Keys that can be used from the Multi-Purpose Keys:**

- `lcc:actionId` mandatory
- `lcc:actionType` mandatory
- `lcc:actionSetId` mandatory
- `lcc:actionColumnValueMinLenToApplyRule` optional
- `lcc:actionColumnValueMaxLenToApplyRule` optional
- `lcc:actionShowSkippedRecords` optional
- `lcc:actionSortMatchAsNumber` optional
- `lcc:actionFlag` optional
- `lcc:actionSkip`  
  optional

**lcc:actionColumnValue (mandatory) (one to many per ASS)**

What column(s) should be used to determine duplicates. These columns must be sorted to correctly find duplicates. You can use the Action 'Sort' just prior to this action to sort the columns.

If you wish to use more than one column to determine duplicates, make sure to sort all of those columns and specify one of these keys for each column.

*Requirement: must be a number*

**Syntax:**

```
lcc:actionColumnValue [tab] [#]
```

**Example:**

```
lcc:actionColumnValue 4
```

**Example:**

```
lcc:actionColumnValue 4
lcc:actionColumnValue 1
lcc:actionColumnValue 2
```

**lcc:actionColumnValue (optional) (one to many per ASS)**

What column(s) should be used to determine duplicates. These columns must be sorted to correctly find duplicates. You can use the Action 'Sort' just prior to this action to sort the columns.

If you wish to use more than one column to determine duplicates, make sure to sort all of those columns and specify one of these keys for each column.

*Requirement: must be a number*

**Syntax:**

```
lcc:actionColumnValue [tab] [#]
```

**Example:**

```
lcc:actionColumnValue 4
```

---

**Example Action Set: Combine Duplicates, Append**

```
lcc:actionId Find Duplicates on SID
lcc:actionType Combine Duplicates, Append
lcc:actionSetId Master
lcc:actionColumnValue 4
lcc:actionColumnValue 1
lcc:actionColumnValue 2
lcc:actionShowSkippedRecords YES
```

---

Lower Columbia College
**Action Keys For: Find Match And Insert**

Additional Keys that can be used from the Multi-Purpose Keys:

- `lcc:actionId` mandatory
- `lcc:actionType` mandatory
- `lcc:actionSetId` mandatory
- `lcc:actionSetId` mandatory
- `lcc:actionColumnValueMinLenToApplyRule` optional
- `lcc:actionColumnValueMaxLenToApplyRule` optional
- `lcc:actionShowSkippedRecords` optional
- `lcc:actionSortMatchAsNumber` optional
- `lcc:actionFlag` optional
- `lcc:actionSkip` optional

**lcc:actionColumnId** (mandatory)(one per ASS)

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax: `lcc:actionColumnId [tab] [ID]`

Example: `lcc:actionColumnId High School Graduate`

**lcc:actionColumnPos** (mandatory)(one per ASS)

The column position number, where '1' is the first column, '2' is second, etc. This column defines the column position from the Data Set you wish to retrieve the value from and the column value to match. If it is a number, see using the key 'lcc:actionSortMatchAsNumber'.

Requirement: must be a number

Syntax: `lcc:actionColumnPos [tab] [#]`

Example: `lcc:actionColumnPos 21`

**lcc:actionColumn2Id** (mandatory)(one per ASS)
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

Syntax: `lcc:actionColumn2Id [tab] [ID]`
Example: `lcc:actionColumn2Id High School Code`

**lcc:actionColumn2Pos (mandatory) (one per ASS)**

The column position number, where '1' is the first column, '2' is second, etc. This column defines the columns position from the Data Set you wish to retrieve the value from and the column value to retrieve.

**Requirement:** must be a number

Syntax: `lcc:actionColumn2Pos [tab] [#]`
Example: `lcc:actionColumn2Pos 21`

**lcc:actionColumn3Id (mandatory) (one per ASS)**

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

Syntax: `lcc:actionColumn3Id [tab] [ID]`
Example: `lcc:actionColumn3Id High School Code`

**lcc:actionColumn3Pos (mandatory) (one per ASS)**

The column position number, where '1' is the first column, '2' is second, etc. This column defines the column position from the Data Set you wish to place the value to and the column value to match. If it is a number, see using the key 'lcc:actionSortMatchAsNumber'.

**Requirement:** must be a number

Syntax: `lcc:actionColumn3Pos [tab] [#]`
Example: `lcc:actionColumn3Pos 21`

**lcc:actionColumn4Id (mandatory) (one per ASS)**

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

Syntax: `lcc:actionColumn4Id [tab] [ID]`
Example: `lcc:actionColumn4Id High School Code`

**lcc:actionColumn4Pos (mandatory) (one per ASS)**

The column position number, where '1' is the first column, '2' is second, etc. This column defines the columns position from the Data Set you wish to place the value to and the column value to insert into that position. A new column will be created.

Lower Columbia College
Requirement: must be a number
Syntax: \texttt{lcc:actionColumn4Pos [tab] [#]}
Example: \texttt{lcc:actionColumn4Pos 21}

\textbf{lcc: actionMatchPreSorted (optional)(one per ASS)}
If set, will assume the columns are sorted. Speeds up the matching formula.

\textbf{Valid values}
- Source

Syntax: \texttt{lcc:actionMatchPreSorted [tab] [YES]}
Example: \texttt{lcc:actionMatchPreSorted YES}

\textbf{lcc: actionMatchHeartbeat (optional)(one per ASS)}
If set, will display/log a heartbeat every \# of records. This is beneficial if you are working with a very large Data Set and want to get periodic feedback as it works to match the records.

Requirement: must be a number
Syntax: \texttt{lcc:actionMatchHeartbeat [tab] [#]}
Example: \texttt{lcc:actionMatchHeartbeat 1000}

Example Action Set: Find Match And Insert
\begin{verbatim}
lcc:actionId Insert Email Address
lcc:actionType Find Match And Insert
lcc:actionSetId lccSingleId Ids
lcc:actionColumnId SID
lcc:actionColumnPos 1
lcc:actionColumn2Id UserId
lcc:actionColumn2Pos 2
lcc:actionSet2Id Master
lcc:actionColumn3Id SID
lcc:actionColumn3Pos 4
lcc:actionColumn4Id Email
lcc:actionColumn4Pos 11
lcc:actionMatchPreSorted YES
\end{verbatim}
lcc:actionMatchHeartbeat  1000
lcc:actionSortMatchAsNumber  YES
lcc:actionSkip  YES

Action Keys For: Find Match And Replace

Additional Keys that can be used from the Multi-Purpose Keys:

- lcc:actionId mandatory
- lcc:actionType mandatory
- lcc:actionSetId mandatory
- lcc:actionSet2Id mandatory
- lcc:actionColumnValueMinLenToApplyRule optional
- lcc:actionColumnValueMaxLenToApplyRule optional
- lcc:actionShowSkippedRecords optional
- lcc:actionSortMatchAsNumber optional
- lcc:actionFlag optional
- lcc:actionSkip optional

lcc:actionColumnId  (mandatory) (one per ASS)
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position
is being affected. Should describe the values in the column.
Syntax:  lcc:actionColumnId [tab] [ID]
Example:  lcc:actionColumnId  Citizenship code

lcc:actionColumnPos  (mandatory) (one per ASS)
The column position number, where '1' is the first column, '2' is second, etc. This column defines the column position
from the Data Set you wish to retrieve the value from and the column value to match. If it is a number, see using the key
'lcc:actionSortMatchAsNumber'.
Requirement: must be a number
Syntax:  lcc:actionColumnPos [tab] [#]
Example:  lcc:actionColumnPos 21

lcc:actionColumn2Id  (mandatory) (one per ASS)

Lower Columbia College
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

Syntax: \texttt{lcc:actionColumn2Id [tab] [ID]}
Example: \texttt{lcc:actionColumn2Id Citizenship code}

\texttt{lcc:actionColumn2Pos (mandatory)(one per ASS)}
The column position number, where '1' is the first column, '2' is second, etc. This column defines the columns position from the Data Set you wish to retrieve the value from and the column value to retrieve.

Requirement: must be a number
Syntax: \texttt{lcc:actionColumn2Pos [tab] [#]}
Example: \texttt{lcc:actionColumn2Pos 21}

\texttt{lcc:actionColumn3Id (mandatory)(one per ASS)}
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

Syntax: \texttt{lcc:actionColumn3Id [tab] [ID]}
Example: \texttt{lcc:actionColumn3Id Citizenship code}

\texttt{lcc:actionColumn3Pos (mandatory)(one per ASS)}
The column position number, where '1' is the first column, '2' is second, etc. This column defines the column position from the Data Set you wish to place the value to and the column value to match. If it is a number, see using the key 'lcc:actionSortMatchAsNumber'.

Requirement: must be a number
Syntax: \texttt{lcc:actionColumn3Pos [tab] [#]}
Example: \texttt{lcc:actionColumn3Pos 21}

\texttt{lcc:actionColumn4Id (mandatory)(one per ASS)}
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

Syntax: \texttt{lcc:actionColumn4Id [tab] [ID]}
Example: \texttt{lcc:actionColumn4Id Citizenship code}

\texttt{lcc:actionColumn4Pos (mandatory)(one per ASS)}
The column position number, where '1' is the first column, '2' is second, etc. This column defines the columns position from the Data Set you wish to place the value to and the column value to replace.

Requirement: must be a number
Syntax: \texttt{lcc:actionColumn4Pos} \texttt{[tab]} \texttt{[#]}
Example: \texttt{lcc:actionColumn4Pos} \texttt{21}

\textbf{\texttt{lcc: actionMatchPreSorted} (optional) (one per ASS)}

If set, will assume the columns are sorted. Speeds up the matching formula.

\textbf{Valid values}
- Source

Syntax: \texttt{lcc:actionMatchPreSorted} \texttt{[tab]} \texttt{[YES]}
Example: \texttt{lcc:actionMatchPreSorted} \texttt{YES}

\textbf{\texttt{lcc: actionMatchHeartbeat} (optional) (one per ASS)}

If set, will display/log a heartbeat every # of records. This is beneficial if you are working with a very large Data Set and want to get periodic feedback as it works to match the records.

\textit{Requirement: must be a number}

Syntax: \texttt{lcc:actionMatchHeartbeat} \texttt{[tab]} \texttt{[#]}
Example: \texttt{lcc:actionMatchHeartbeat} \texttt{1000}

Example Action Set: Find Match And Replace

\texttt{lcc:actionId} \texttt{Find Match On Citizenship status And Replace Citizenship Status}
\texttt{lcc:actionType} \texttt{Find Match And Replace}
\texttt{lcc:actionSetId} \texttt{Citizenship Codes}
\texttt{lcc:actionColumnId} \texttt{Code}
\texttt{lcc:actionColumnPos} \texttt{1}
\texttt{lcc:actionColumn2Id} \texttt{Title}
\texttt{lcc:actionColumn2Pos} \texttt{2}
\texttt{lcc:actionSet2Id} \texttt{Master}
\texttt{lcc:actionColumn3Id} \texttt{Citizenship code}
\texttt{lcc:actionColumn3Pos} \texttt{14}
\texttt{lcc:actionColumn4Id} \texttt{Citizenship code}
\texttt{lcc:actionColumn4Pos} \texttt{14}
\texttt{lcc:actionMatchPreSorted} \texttt{YES}
\texttt{lcc:actionMatchHeartbeat} \texttt{1000}
**Action Keys For: If Empty, Place Value**

Additional Keys that can be used from the Multi-Purpose Keys:
- `lcc:actionId` *mandatory*
- `lcc:actionType` *mandatory*
- `lcc:actionSetId` *mandatory*
- `lcc:actionColumnValueMinLenToApplyRule` *optional*
- `lcc:actionColumnValueMaxLengthToApplyRule` *optional*
- `lcc:actionFlag` *optional*
- `lcc:actionSkip` *optional*

**lcc:actionColumnId** *(mandatory) (one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

*Syntax:* `lcc:actionColumnId [tab] [ID]`

*Example:* `lcc:actionColumnId Last Name`

**lcc:actionColumnPos** *(mandatory) (one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column will be checked to see if it empty.

*Requirement:* must be a number

*Syntax:* `lcc:actionColumnPos [tab] [#]`

*Example:* `lcc:actionColumnPos 1`

**lcc:actionColumn2Id** *(mandatory) (one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

*Syntax:* `lcc:actionColumn2Id [tab] [ID]`

*Example:* `lcc:actionColumn2Id First Name`

**lcc:actionColumn2Pos** *(mandatory) (one per ASS)*

Lower Columbia College
The column position number, where '1' is the first column, '2' is second, etc. What column will receive a value.

Requirement: must be a number
Syntax: \texttt{lcc:actionColumn2Pos \texttt{tab} \#}
Example: \texttt{lcc:actionColumn2Pos 2}

Example Action Set: If Empty, Place Value
\begin{verbatim}
\texttt{lcc:actionId Where Last Name Is Empty}
\texttt{lcc:actionType If Empty, Place Value}
\texttt{lcc:actionSetId Master}
\texttt{lcc:actionColumnId Last Name}
\texttt{lcc:actionColumnPos 1}
\texttt{lcc:actionColumn2Id Notes}
\texttt{lcc:actionColumn2Pos 6}
\texttt{lcc:actionColumnValue Last name was empty.}
\texttt{lcc:actionSkip YES}
\end{verbatim}

Action Keys For: If Not Empty, Place Value

Additional Keys that can be used from the Multi-Purpose Keys:
- \texttt{lcc:actionId} \textit{mandatory}
- \texttt{lcc:actionType} \textit{mandatory}
- \texttt{lcc:actionSetId} \textit{mandatory}
- \texttt{lcc:actionColumnValueMinLenToApplyRule} \textit{optional}
- \texttt{lcc:actionColumnValueMaxLenToApplyRule} \textit{optional}
- \texttt{lcc:actionFlag} \textit{optional}
- \texttt{lcc:actionSkip} \textit{optional}

\texttt{lcc:actionColumnId} \textit{(mandatory) (one per ASS)}
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.
Syntax: \texttt{lcc:actionColumnId \texttt{tab} [ID]}
Example: \texttt{lcc:actionColumnId Last Name}
**lcc:actionColumnPos** *(mandatory) (one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column will be checked to see if it not empty.

*Requirement: must be a number*

**Syntax:** 
```
  lcc:actionColumnPos [tab] [#]
```

**Example:**
```
  lcc:actionColumnPos 1
```

**lcc:actionColumn2Id** *(mandatory) (one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

**Syntax:**
```
  lcc:actionColumn2Id [tab] [ID]
```

**Example:**
```
  lcc:actionColumn2Id First Name
```

**lcc:actionColumn2Pos** *(mandatory) (one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column will receive a value.

*Requirement: must be a number*

**Syntax:**
```
  lcc:actionColumn2Pos [tab] [#]
```

**Example:**
```
  lcc:actionColumn2Pos 2
```

**Example Action Set: If Not Empty, Place Value**
```
lcc:actionId    Where Last Name Is Not Empty
lcc:actionType   If Not Empty, Place Value
lcc:actionSetId    Master
lcc:actionColumnId   Last Name
lcc:actionColumnPos 1
lcc:actionColumn2Id Notes
lcc:actionColumn2Pos   6
lcc:actionColumnValue   Last name was not empty.
  lcc:actionSkip YES
```

**Action Keys For: Insert Empty Columns**

**Additional Keys that can be used from the Multi-Purpose Keys:**

---

Lower Columbia College
- **lcc:actionId** (mandatory)
- **lcc:actionType** (mandatory)
- **lcc:actionSetId** (mandatory)
- **lcc:actionFlag** (optional)
- **lcc:actionSkip** (optional)

**lcc:actionColumnId** (mandatory) (one per ASS)

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax: `lcc:actionColumnId [tab] [ID]
Example: `lcc:actionColumnId Enrollment Level`

**lcc:actionColumnPos** (mandatory) (one per ASS)

The column position number, where '1' is the first column, '2' is second, etc. This column defines the column position where empty column(s) will be inserted.

*Requirement: must be a number*

Syntax: `lcc:actionColumnPos [tab] [#]
Example: `lcc:actionColumnPos 18`

**lcc:actionColumnValue** (mandatory) (one per ASS)

How many columns to insert.

Syntax: `lcc:actionColumnValue [tab] [...]`
Example: `lcc:actionColumnValue 1`

Example Action Set: Insert Empty Columns

- **lcc:actionId** Insert Column for Enrollment Level
- **lcc:actionType** Insert Empty Columns
- **lcc:actionSetId** Master
- **lcc:actionColumnId** Enrollment Level
- **lcc:actionColumnPos 18
- **lcc:actionColumnValue 1
  - **lcc:actionSkip** YES

Lower Columbia College
Action Keys For: Insert Line At Each Record Set

Additional Keys that can be used from the Multi-Purpose Keys:

- **lcc:actionId** mandatory
- **lcc:actionType** mandatory
- **lcc:actionSetId** mandatory
- **lcc:actionColumnValueMinLenToApplyRule** optional
- **lcc:actionColumnValueMaxLenToApplyRule** optional
- **lcc:actionShowSkippedRecords** optional
- **lcc:actionSkip** optional

**lcc:actionColumnId** *(mandatory)* *(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax:  

```
lcc:actionColumnId [tab] [ID]
```

Example:

```
lcc:actionColumnId Quarter Credits
```

**lcc:actionColumnPos** *(mandatory)* *(one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column to use when finding Record Sets. Each time this column value changes, a new record set is assumed. Recommend sorting the records by this column first.

*Requirement: must be a number*

Syntax:  

```
lcc:actionColumnPos [tab] [#]
```

Example:

```
lcc:actionColumnPos 17
```

**lcc:actionValue** *(mandatory)* *(one per ASS)*

What line will be placed after each record set.

You can also specify these white-space characters:

- `[tab]`: will change to a single tab character with processing
- `[space]`: will change to a single space character with processing

Syntax:  

```
lcc:actionValue [tab] [...]`

Lower Columbia College
Example:

```
  lcc:actionValue   *** a new record set starts here ***
```

Example Action Set: Insert Line At Each Record Set

- **lcc:actionId**: Record Set Separator
- **lcc:actionType**: Insert Line At Each Record Set
- **lcc:actionSetId**: Master
- **lcc:actionColumnId**: SSN
- **lcc:actionColumnPos**: 25
- **lcc:actionValue**: *** a new record set starts here ***
- **lcc:actionSkip**: YES

**Action Keys For: Keep Records With Column Value**

Additional Keys that can be used from the Multi-Purpose Keys:

- **lcc:actionId**: mandatory
- **lcc:actionType**: mandatory
- **lcc:actionSetId**: mandatory
- **lcc:actionFlag**: optional
- **lcc:actionSkip**: optional

**lcc:actionColumnId** *(mandatory)*(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

**Syntax:**

```
  lcc:actionColumnId [tab] [ID]
```

**Example:**

```
  lcc:actionColumnId   High School Graduate
```

**lcc:actionColumnPos** *(mandatory)*(one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column value to use as a comparison against the supplied values. Records that match any of the supplied values will be kept, all other will be skipped.

**Requirement:** must be a number

**Syntax:**

```
  lcc:actionColumnPos [tab] [#]
```

**Example:**

```
  lcc:actionColumnPos 21
```
**lcc:actionColumnValue** *(mandatory) (one to many per ASS)*

What values to compare. If more than one, supply this key for each value.

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

Example: `lcc:actionColumnValue 1`

Example: `lcc:actionColumnValue Yes`

Example: `lcc:actionColumnValue Maybe`

Example Action Set: Keep Records With Column Value

lcc:actionId Keep Records Where HS Grad is Y
lcc:actionType Keep Records With Column Value
lcc:actionSetId Master
lcc:actionColumnId High School Graduate
lcc:actionColumnPos 21
lcc:actionColumnValue Y
lcc:actionSkip YES

**Action Keys For: Keep Records With Column Value Containing**

Additional Keys that can be used from the Multi-Purpose Keys:

- `lcc:actionId` *mandatory*
- `lcc:actionType` *mandatory*
- `lcc:actionSetId` *mandatory*
- `lcc:actionFlag` *optional*
- `lcc:actionSkip` *optional*

**lcc:actionColumnInfo** *(mandatory) (one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax: `lcc:actionColumnInfo [tab] [ID]`

Example: `lcc:actionColumnInfo High School Graduate`

**lcc:actionColumnPos** *(mandatory) (one per ASS)*

Lower Columbia College
The column position number, where '1' is the first column, '2' is second, etc. What column value to use as a comparison against the supplied values. Records that contain any of the supplied values will be kept, all other will be skipped.

**Requirement:** must be a number

**Syntax:**
```
   lcc:actionColumnPos {tab} [\#]
```

**Example:**
```
lcc:actionColumnPos 21
```

**lcc:actionColumnValue** *(mandatory)* *(one to many per ASS)*

What values to compare. If more than one, supply this key for each value.

**Syntax:**
```
   lcc:actionColumnValue {tab} [...] 
```

**Example:**
```
lcc:actionColumnValue 1
```
```
lcc:actionColumnValue Yes
```
```
lcc:actionColumnValue Maybe
```

**Example Action Set:** Keep Records With Column Value Containing
```
lcc:actionId   Keep Records Where EFC contains .00
lcc:actionType Keep Records With Column Value Containing
lcc:actionSetId Master
lcc:actionColumnId EFC
lcc:actionColumnPos 1
lcc:actionColumnValue .00
   lcc:actionSkip YES
```

**Action Keys For: Keep Records With Empty Column**

Additional Keys that can be used from the Multi-Purpose Keys:
- `lcc:actionId` *mandatory*
- `lcc:actionType` *mandatory*
- `lcc:actionSetId` *mandatory*
- `lcc:actionColumnValueMinLenToApplyRule` *optional*
- `lcc:actionColumnValueMaxLenToApplyRule` *optional*
- `lcc:actionFlag` *optional*
- `lcc:actionSkip` *optional*

**lcc:actionColumnId** *(mandatory)* *(one per ASS)*
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax: `lcc:actionColumnId [tab] [ID]`
Example: `lcc:actionColumnId Last Name`

**lcc:actionColumnPos (mandatory)(one per ASS)**

The column position number, where '1' is the first column, '2' is second, etc. What column will be checked to see if it is empty.

Requirement: must be a number

Syntax: `lcc:actionColumnPos [tab] [#]`
Example: `lcc:actionColumnPos 1`

Example Action Set: Keep Records With Empty Column

```
lcc:actionId Where EFC is empty
lcc:actionType Keep Records With Empty Column
lcc:actionSetId Master
lcc:actionColumnId EFC
lcc:actionColumnPos 1
lcc:actionSkip YES
```

**Action Keys For: Keep Records With Not Empty Column**

Additional Keys that can be used from the Multi-Purpose Keys:

- `lcc:actionId` mandatory
- `lcc:actionType` mandatory
- `lcc:actionSetId` mandatory
- `lcc:actionColumnValueMinLenToApplyRule` optional
- `lcc:actionColumnValueMaxLenToApplyRule` optional
- `lcc:actionFlag` optional
- `lcc:actionSkip` optional

**lcc:actionColumnId (mandatory)(one per ASS)**

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.
Syntax:  lcc:actionColumnId [tab] [ID]
Example:  lcc:actionColumnId  Last Name

**lcc:actionColumnPos** *(mandatory)* *(one per ASS)*
The column position number, where '1' is the first column, '2' is second, etc. What column will be checked to see if it is not empty.
*Requirement: must be a number*
Syntax:  lcc:actionColumnPos [tab] [#]
Example:  lcc:actionColumnPos 1

Example Action Set: Keep Records With Not Empty Column
lcc:actionId  Where EFC is not empty
lcc:actionType  Keep Records With Not Empty Column
lcc:actionSetId  Master
lcc:actionColumnId  EFC
lcc:actionColumnPos 1
lcc:actionSkip YES

**Action Keys For: Lower Case**

Additional Keys that can be used from the Multi-Purpose Keys:
- lcc:actionId  *mandatory*
- lcc:actionType  *mandatory*
- lcc:actionSetId  *mandatory*
- lcc:actionColumnValueMinLenToApplyRule  *optional*
- lcc:actionColumnValueMaxLenToApplyRule  *optional*
- lcc:actionFlag  *optional*
- lcc:actionSkip  *optional*

**lcc:actionColumnId** *(mandatory)* *(one per ASS)*
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.
Syntax:  lcc:actionColumnId [tab] [ID]
Example:  lcc:actionColumnId  Street

Lower Columbia College
**lcc:actionColumnPos** *(mandatory) (one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. This value will be lower cased.

*Requirement:* must be a number

*Syntax:*   
lcc:actionColumnPos [tab] [#]

*Example:*   
lcc:actionColumnPos 7

**Example Action Set: Lower Case**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>lcc:actionId</td>
<td>Lower Case - Street</td>
</tr>
<tr>
<td>lcc:actionType</td>
<td>Lower Case</td>
</tr>
<tr>
<td>lcc:actionSetId</td>
<td>Master</td>
</tr>
<tr>
<td>lcc:actionColumnId</td>
<td>Street</td>
</tr>
<tr>
<td>lcc:actionColumnPos</td>
<td>7</td>
</tr>
<tr>
<td>lcc:actionSkip</td>
<td>YES</td>
</tr>
</tbody>
</table>

**Action Keys For: Math: if <=, replace**

Additional Keys that can be used from the Multi-Purpose Keys:

- **lcc:actionId**  
  *mandatory*
- **lcc:actionType**  
  *mandatory*
- **lcc:actionSetId**  
  *mandatory*
- **lcc:actionFlag**  
  *optional*
- **lcc:actionSkip**  
  *optional*

**lcc:actionColumnId** *(mandatory) (one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

*Syntax:*   
lcc:actionColumnId [tab] [ID]

*Example:*   
lcc:actionColumnId Quarter Credits

**lcc:actionColumnPos** *(mandatory) (one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column value to compare against, must be lesser or equal to comparison.

Lower Columbia College
**Requirement:** must be a number

Syntax: \lcc\:actionColumnPos \{tab\} \[#\]

Example: \lcc\:actionColumnPos 21

**lcc\:actionColumn2Id** *(mandatory) (one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

Syntax: \lcc\:actionColumn2Id \{tab\} \[ID\]

Example: \lcc\:actionColumn2Id Enrollment Level

**lcc\:actionColumn2Pos** *(mandatory) (one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column value should be replaced.

Requirement: must be a number

Syntax: \lcc\:actionColumn2Pos \{tab\} \[#\]

Example: \lcc\:actionColumn2Pos 21

**lcc\:actionColumnValue** *(mandatory) (one per ASS)*

What number this value should be less or equal to.

Syntax: \lcc\:actionColumnValue \{tab\} \[#\]

Example: \lcc\:actionColumnValue 0

**lcc\:actionValue** *(mandatory) (one per ASS)*

What value to use for replacement.

Syntax: \lcc\:actionValue \{tab\} \[...\]

Example: \lcc\:actionValue not enrolled

Example Action Set: Math: if <=, replace

- \lcc\:actionId Convert Enrollment Level to Title, if <= 0
- \lcc\:actionType Math: if <=, replace
- \lcc\:actionSetId Master
- \lcc\:actionColumnId Quarter Credits
- \lcc\:actionColumnPos 17
- \lcc\:actionColumn2Id Enrollment Level
- \lcc\:actionColumn2Pos 18
lcc:actionColumnValue  0
lcc:actionValue not enrolled
lcc:actionSkip YES

Action Keys For: Math: if >= and <=, replace

Additional Keys that can be used from the Multi-Purpose Keys:
- lcc:actionId  mandatory
- lcc:actionType  mandatory
- lcc:actionSetId  mandatory
- lcc:actionFlag  optional
- lcc:actionSkip  optional

lcc:actionColumnId  (mandatory)(one per ASS)
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.
Syntax: lcc:actionColumnId [tab] [ID]
Example: lcc:actionColumnId Quarter Credits

lcc:actionColumnPos  (mandatory)(one per ASS)
The column position number, where '1' is the first column, '2' is second, etc. What column value to compare against, must be greater or equal to and lesser or equal to comparison.
Requirement: must be a number
Syntax: lcc:actionColumnPos [tab] [#]
Example: lcc:actionColumnPos 17

lcc:actionColumn2Id  (mandatory)(one per ASS)
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.
Syntax: lcc:actionColumn2Id [tab] [ID]
Example: lcc:actionColumn2Id Enrollment Level

lcc:actionColumn2Pos  (mandatory)(one per ASS)
The column position number, where '1' is the first column, '2' is second, etc. What column value should be replaced.
Lower Columbia College
Action Keys For: Math: if >=, replace

Additional Keys that can be used from the Multi-Purpose Keys:

- **lcc:actionId** mandatory
- **lcc:actionType** mandatory
- **lcc:actionSetId** mandatory
- **lcc:actionFlag** optional
- **lcc:actionSkip** optional

**lcc:actionColumnId** (mandatory) (one per ASS)

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax: `lcc:actionColumnId [tab] [ID]`
Example: `lcc:actionColumnId Quarter Credits`

**lcc:actionColumnPos** (mandatory) (one per ASS)

The column position number, where '1' is the first column, '2' is second, etc. What column value to compare against, must be greater or equal to comparison.

Requirement: must be a number
Syntax: `lcc:actionColumnPos [tab] [#]`
Example: `lcc:actionColumnPos 17`

**lcc:actionColumn2Id** (mandatory) (one per ASS)

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

Syntax: `lcc:actionColumn2Id [tab] [ID]`
Example: `lcc:actionColumn2Id Enrollment Level`

**lcc:actionColumn2Pos** (mandatory) (one per ASS)

The column position number, where '1' is the first column, '2' is second, etc. What column value should be replaced.

Requirement: must be a number
Syntax: `lcc:actionColumn2Pos [tab] [#]`
Example: `lcc:actionColumn2Pos 21`
**lcc:actionColumnValue** *(mandatory)(one per ASS)*

*Requirement:* must be a number

What number this value should be greater than or equal to.

*Syntax:*  
```
lcc:actionColumnValue [tab] [#]
```

*Example:*  
```
lcc:actionColumnValue 1
```

**lcc:actionValue** *(mandatory)(one per ASS)*

What value to use for replacement.

*Syntax:*  
```
lcc:actionValue [tab] [...]  
```

*Example:*  
```
lcc:actionValue full time  
```

Example Action Set: Math: if >=, replace
```
lcc:actionId Convert Enrollment Level to Title, if >=12  
lcc:actionType Math: if >=, replace  
lcc:actionSetId Master  
lcc:actionColumnId Quarter Credits  
lcc:actionColumnPos 17  
lcc:actionColumn2Id Enrollment Level  
lcc:actionColumn2Pos 18  
lcc:actionColumnValue 12  
lcc:actionValue full time  
lcc:actionSkip YES
```

**Action Keys For:** Math: Multiply

Additional Keys that can be used from the Multi-Purpose Keys:
- **lcc:actionId** *mandatory*
- **lcc:actionType** *mandatory*
- **lcc:actionSetId** *mandatory*
- **lcc:actionFlag** *optional*
- **lcc:actionSkip** *optional*
**lcc:actionColumnId** *(mandatory)* *(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax:  
```
    lcc:actionColumnId [tab] [ID]
```

Example:
```
    lcc:actionColumnId  Quarter Credits
```

**lcc:actionColumnPos** *(mandatory)* *(one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column value to multiply. The value will then be replaced with the new number.

*Requirement:* must be a number

Syntax:  
```
    lcc:actionColumnPos [tab] [#]
```

Example:
```
    lcc:actionColumnPos 17
```

**lcc:actionColumnValue** *(mandatory)* *(one per ASS)*

*Requirement:* must be a number

What is the multiplier.

Syntax:  
```
    lcc:actionColumnValue [tab] [#]
```

Example:
```
    lcc:actionColumnValue .01
```

Example Action Set: Math: Multiply

```
lcc:actionId Multiply GPA by .01
lcc:actionType Math: Multiply
lcc:actionSetId Master
lcc:actionColumnId GPA
lcc:actionColumnPos 15
lcc:actionColumnValue .01
lcc:actionSkip YES
```

**Action Keys For: Pad Left**

*Additional Keys that can be used from the Multi-Purpose Keys:*

- **lcc:actionId** mandatory
- **lcc:actionType** mandatory
- **lcc:actionSetId** mandatory
lcc:actionColumnId (mandatory)(one per ASS)
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.
Syntax: lcc:actionColumnId [tab] [ID]
Example: lcc:actionColumnId Quarter Credits

lcc:actionColumnPos (mandatory)(one per ASS)
The column position number, where '1' is the first column, '2' is second, etc. What column value to left pad with a character.
Requirement: must be a number
Syntax: lcc:actionColumnPos [tab] [#]
Example: lcc:actionColumnPos 17

lcc:actionColumnValue (mandatory)(one per ASS)
What character to use as the left pad.

You can also specify these white-space characters:
- [tab]: will change to a single tab character with processing
- [space]: will change to a single space character with processing

Requirement: must be a single character, or a White Space key word.
Syntax: lcc:actionColumnValue [tab] [#]
Example: lcc:actionColumnValue .01

lcc:actionValue (mandatory)(one per ASS)
What length of the value should be left padded up to. For example, if you put '9' and the value is only 5 characters long, it will pad 4 characters on the left to make the length 9.
Syntax: lcc:actionValue [tab] [...]
Example Action Set: Pad Left

**lcc:actionId**  Left pad 0 To Master SSN  
**lcc:actionType**  Pad Left  
**lcc:actionSetId**  Master  
**lcc:actionColumnId**  SSN  
**lcc:actionColumnPos**  25  
**lcc:actionColumnValue**  0  
**lcc:actionValue**  9  
**lcc:actionColumnValueMinLenToApplyRule**  1  
**lcc:actionSkip**  YES

**Action Keys For: Pad Right**

Additional Keys that can be used from the Multi-Purpose Keys:
- **lcc:actionId**  mandatory
- **lcc:actionType**  mandatory
- **lcc:actionSetId**  mandatory
- **lcc:actionColumnValueMinLenToApplyRule**  optional
- **lcc:actionColumnValueMaxLenToApplyRule**  optional
- **lcc:actionShowSkippedRecords**  optional
- **lcc:actionFlag**  optional
- **lcc:actionSkip**  optional

**lcc:actionColumnId**  *(mandatory)* *(one per ASS)*  
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

**Syntax:**  `lcc:actionColumnId [tab] [ID]`

**Example:**  `lcc:actionColumnId  Quarter Credits`

**lcc:actionColumnPos**  *(mandatory)* *(one per ASS)*
The column position number, where '1' is the first column, '2' is second, etc. What column value to right pad with a character.

**Requirement:** must be a number

**Syntax:**  
```
 lcc:actionColumnPos [tab] [#]
```

**Example:**  
```
 lcc:actionColumnPos 17
```

**lcc:actionColumnValue** *(mandatory)* *(one per ASS)*

What character to use as the right pad.

You can also specify these white-space characters:

- `[tab]`: will change to a single tab character with processing
- `[space]`: will change to a single space character with processing

**Requirement:** must be a single character, or a White Space key word.

**Syntax:**  
```
 lcc:actionColumnValue [tab] [#]
```

**Example:**  
```
 lcc:actionColumnValue .01
```

**lcc:actionValue** *(mandatory)* *(one per ASS)*

What length of the value should be right padded up to. For example, if you put '9' and the value is only 5 characters long, it will pad 4 characters on the right to make the length 9.

**Syntax:**  
```
 lcc:actionValue [tab] [...]
```

**Example:**  
```
 lcc:actionValue not enrolled
```

Example Action Set: Pad Right

```
 lcc:actionId     Left pad 0 To Master SSN
 lcc:actionType   Pad Right
 lcc:actionSetId  Master
 lcc:actionColumnId SSN
 lcc:actionColumnPos 25
 lcc:actionColumnValue     0
 lcc:actionValue          9
 lcc:actionColumnValueMinLenToApplyRule 1
 lcc:actionSkip        YES
```

Lower Columbia College
Action Keys For: Prepend

Additional Keys that can be used from the Multi-Purpose Keys:

- `lcc:actionId` mandatory
- `lcc:actionType` mandatory
- `lcc:actionSetId` mandatory
- `lcc:actionColumnValueMinLenToApplyRule` optional
- `lcc:actionColumnValueMaxLenToApplyRule` optional
- `lcc:actionShowSkippedRecords` optional
- `lcc:actionFlag` optional
- `lcc:actionSkip` optional

`lcc:actionColumnId` *(mandatory)* *(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

**Syntax:** `lcc:actionColumnId [tab] [ID]`

**Example:** `lcc:actionColumnId DOB`

`lcc:actionColumnPos` *(mandatory)* *(one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column value to have its value prepended to.

**Requirement:** must be a number

**Syntax:** `lcc:actionColumnPos [tab] [#]`

**Example:** `lcc:actionColumnPos 5`

`lcc:actionColumnValue` *(mandatory)* *(one per ASS)*

What value to use to prepend.

**Syntax:** `lcc:actionColumnValue [tab] [...]`

**Example:** `lcc:actionColumnValue 0`
Example Action Set: Prepend

<table>
<thead>
<tr>
<th>lcc:actionId</th>
<th>Prepend 0 to DOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>lcc:actionType</td>
<td>Prepend</td>
</tr>
<tr>
<td>lcc:actionSetId</td>
<td>Master</td>
</tr>
<tr>
<td>lcc:actionColumnId</td>
<td>DOB</td>
</tr>
<tr>
<td>lcc:actionColumnPos</td>
<td>5</td>
</tr>
<tr>
<td>lcc:actionColumnValue</td>
<td>0</td>
</tr>
<tr>
<td>lcc:actionColumnValueMaxLenToApplyRule</td>
<td>6</td>
</tr>
<tr>
<td>lcc:actionSkip</td>
<td>YES</td>
</tr>
</tbody>
</table>

Action Keys For: Remove Columns

Additional Keys that can be used from the Multi-Purpose Keys:

- lcc:actionId \textit{mandatory}
- lcc:actionType \textit{mandatory}
- lcc:actionSetId \textit{mandatory}
- lcc:actionFlag \textit{optional}
- lcc:actionSkip \textit{optional}

\textbf{lcc:actionColumnId} \textit{(mandatory)(one per ASS)}

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

\textbf{Syntax:} \texttt{lcc:actionColumnId [tab] [ID]}

\textbf{Example:} \texttt{lcc:actionColumnId Enrollment Level}

\textbf{lcc:actionColumnPos} \textit{(mandatory)(one per ASS)}

The column position number, where '1' is the first column, '2' is second, etc. This column defines the column position where column(s) will be removed.

\textit{Requirement: must be a number}

\textbf{Syntax:} \texttt{lcc:actionColumnPos [tab] [#]}

Lower Columbia College
Example:    lcc:actionColumnPos 18

**lcc:actionColumnValue** *(mandatory)* *(one per ASS)*
How many columns to remove.
Syntax:    lcc:actionColumnValue [tab] [...] 
Example:    lcc:actionColumnValue 1

Example Action Set: Remove Columns
lcc:actionId    Remove Columns for Program Title and Program Type
lcc:actionType    Remove Columns
lcc:actionSetId    Master
lcc:actionColumnId    Program Title and Program Type
lcc:actionColumnPos 23
lcc:actionColumnValue 2
    lcc:actionSkip YES

**Action Keys For: Remove Consecutive Spaces**

Additional Keys that can be used from the Multi-Purpose Keys:
- **lcc:actionId** *(mandatory)*
- **lcc:actionType** *(mandatory)*
- **lcc:actionSetId** *(mandatory)*
- **lcc:actionColumnValueMinLenToApplyRule** *(optional)*
- **lcc:actionColumnValueMaxLenToApplyRule** *(optional)*
- **lcc:actionFlag** *(optional)*
- **lcc:actionSkip** *(optional)*

**lcc:actionColumnId** *(mandatory)* *(one per ASS)*
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.
Syntax:    lcc:actionColumnId [tab] [ID]
Example:    lcc:actionColumnId  Street
**lcc:actionColumnPos** *(mandatory)*(one per ASS)

The column position number, where '1' is the first column, '2' is second, etc. This value will have consecutive spaces removed when there is more than one together.

*Requirement: must be a number*

*Syntax:*  
lcc:actionColumnPos [tab] [#]

*Example:*  
lcc:actionColumnPos 7

---

**Example Action Set: Remove Consecutive Spaces**

<table>
<thead>
<tr>
<th>lcc:actionId</th>
<th>Remove Consecutive Spaces - Street</th>
</tr>
</thead>
<tbody>
<tr>
<td>lcc:actionType</td>
<td>Remove Consecutive Spaces</td>
</tr>
<tr>
<td>lcc:actionSetId</td>
<td>Master</td>
</tr>
<tr>
<td>lcc:actionColumnId</td>
<td>Street</td>
</tr>
<tr>
<td>lcc:actionColumnPos</td>
<td>7</td>
</tr>
<tr>
<td>lcc:actionSkip</td>
<td>YES</td>
</tr>
</tbody>
</table>

---

**Action Keys For: Replace**

Additional Keys that can be used from the Multi-Purpose Keys:

- **lcc:actionId** *(mandatory)*
- **lcc:actionType** *(mandatory)*
- **lcc:actionSetId** *(mandatory)*
- **lcc:actionColumnValueMinLenToApplyRule** *(optional)*
- **lcc:actionColumnValueMaxLenToApplyRule** *(optional)*
- **lcc:actionFlag** *(optional)*
- **lcc:actionSkip** *(optional)*

**lcc:actionColumnId** *(mandatory)*(one per ASS)

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

*Syntax:*  
lcc:actionColumnId [tab] [ID]

*Example:*  
lcc:actionColumnId  Email
The column position number, where '1' is the first column, '2' is second, etc. Parts of this value will be replaced.

**Requirement:** must be a number

**Syntax:**  `lcc:actionColumnPos [tab] [#]`

**Example:**  `lcc:actionColumnPos 11`

**lcc:actionColumnValue (mandatory) (one per ASS)**

What value should be searched for in the column value. This should be first one supplied.

**Syntax:**  `lcc:actionColumnValue [tab] [...]`

**Example:**  `lcc:actionColumnValue dog`

**lcc:actionColumnValue (mandatory) (one per ASS)**

What value should be used to replace any matched values. This should be the second one supplied.

**Syntax:**  `lcc:actionColumnValue [tab] [...]`

**Example:**  `lcc:actionColumnValue cat`

---

**Example Action Set: Replace**

```
lcc:actionId   Replace dog with cat
lcc:actionType Replace
lcc:actionSetId Master
lcc:actionColumnId Animal
lcc:actionColumnPos 11
lcc:actionColumnValue   dog
lcc:actionColumnValue   cat
```

**Action Keys For: Skip Records With Column Value**

**Additional Keys that can be used from the Multi-Purpose Keys:**

- `lcc:actionId`  *mandatory*
- `lcc:actionType`  *mandatory*
- `lcc:actionSetId`  *mandatory*
- `lcc:actionFlag`  *optional*
- `lcc:actionSkip`  *optional*
**lcc:actionColumnId** *(mandatory)(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

**Syntax:**  
```plaintext
lcc:actionColumnId [tab] [ID]
```

**Example:**  
```plaintext
lcc:actionColumnId High School Graduate
```

**lcc:actionColumnPos** *(mandatory)(one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column value to use as a comparison against the supplied values. Records that do not match any of the supplied values will be skipped, all other will be kept.

**Requirement:** must be a number

**Syntax:**  
```plaintext
lcc:actionColumnPos [tab] [#]
```

**Example:**  
```plaintext
lcc:actionColumnPos 21
```

**lcc:actionColumnValue** *(mandatory)(one to many per ASS)*

What values to compare. If more than one, supply this key for each value.

**Syntax:**  
```plaintext
lcc:actionColumnValue [tab] [...]
```

**Example:**  
```plaintext
lcc:actionColumnValue 1
lcc:actionColumnValue Yes
lcc:actionColumnValue Maybe
```

Example Action Set: Skip Records With Column Value

- `lcc:actionId` Skip Records Where HS Grad is Y
- `lcc:actionType` Skip Records With Column Value
- `lcc:actionSetId` Master
- `lcc:actionColumnId` High School Graduate
- `lcc:actionColumnPos 21`
- `lcc:actionColumnValue No`
- `lcc:actionSkip YES`

**Action Keys For: Skip Records With Column Value Containing**

Additional Keys that can be used from the Multi-Purpose Keys:
- **lcc:actionId** mandatory
- **lcc:actionType** mandatory
- **lcc:actionSetId** mandatory
- **lcc:actionFlag** optional
- **lcc:actionSkip** optional

### lcc:actionColumnId (mandatory)(one per ASS)
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

**Syntax:** `lcc:actionColumnId [tab] [ID]`

**Example:** `lcc:actionColumnId High School Graduate`

### lcc:actionColumnPos (mandatory)(one per ASS)
The column position number, where '1' is the first column, '2' is second, etc. What column value to use as a comparison against the supplied values. Records that do not contain any of the supplied values will be skipped, all other will be kept.

**Requirement:** must be a number

**Syntax:** `lcc:actionColumnPos [tab] [#]`

**Example:** `lcc:actionColumnPos 21`

### lcc:actionColumnValue (mandatory)(one to many per ASS)
What values to compare. If more than one, supply this key for each value.

**Syntax:** `lcc:actionColumnValue [tab] [...]`

**Example:**
- `lcc:actionColumnValue 1`
- `lcc:actionColumnValue Yes`
- `lcc:actionColumnValue Maybe`

**Example Action Set:** Skip Records With Column Value Containing
```
lcc:actionId    Skip Records Where EFC contains .00
lcc:actionType  Skip Records With Column Value Containing
lcc:actionSetId Master
lcc:actionColumnId EFC
lcc:actionColumnPos 1
lcc:actionColumnValue .00
```

Lower Columbia College
lcc:actionSkip YES

**Action Keys For: Skip Records With Empty Column**

Additional Keys that can be used from the Multi-Purpose Keys:
- `lcc:actionId` mandatory
- `lcc:actionType` mandatory
- `lcc:actionSetId` mandatory
- `lcc:actionColumnValueMinLenToApplyRule` optional
- `lcc:actionColumnValueMaxLenToApplyRule` optional
- `lcc:actionFlag` optional
- `lcc:actionSkip` optional

`lcc:actionColumnId` *(mandatory)* *(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax: `lcc:actionColumnId [tab] [ID]`

Example: `lcc:actionColumnId Last Name`

`lcc:actionColumnPos` *(mandatory)* *(one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column will be checked to see if it is empty.

*Requirement: must be a number*

Syntax: `lcc:actionColumnPos [tab] [#]`

Example: `lcc:actionColumnPos 1`

Example Action Set: Skip Records With Empty Column

```
lcc:actionId   Where EFC is empty
lcc:actionType Skip Records With Empty Column
lcc:actionSetId Master
lcc:actionColumnId EFC
lcc:actionColumnPos 1
lcc:actionSkip YES
```

Lower Columbia College
Action Keys For: Skip Records With Not Empty Column

Additional Keys that can be used from the Multi-Purpose Keys:

- **lcc:actionId**  *mandatory*
- **lcc:actionType**  *mandatory*
- **lcc:actionSetId**  *mandatory*
- **lcc:actionColumnValueMinLenToApplyRule**  *optional*
- **lcc:actionColumnValueMaxLenToApplyRule**  *optional*
- **lcc:actionFlag**  *optional*
- **lcc:actionSkip**  *optional*

**lcc:actionColumnId** *(mandatory) (one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

*Syntax:*  `lcc:actionColumnId [tab] [ID]`

*Example:*  `lcc:actionColumnId Last Name`

**lcc:actionColumnPos** *(mandatory) (one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column will be checked to see if it is not empty.

*Requirement:*  *must be a number*

*Syntax:*  `lcc:actionColumnPos [tab] [#]`

*Example:*  `lcc:actionColumnPos 1`

Example Action Set: Skip Records With Not Empty Column

- **lcc:actionId**  *Where EFC is not empty*
- **lcc:actionType**  *Skip Records With Not Empty Column*
- **lcc:actionSetId**  *Master*
- **lcc:actionColumnId**  *EFC*
- **lcc:actionColumnPos 1**
- **lcc:actionSkip YES**
Action Keys For: Sort

Additional Keys that can be used from the Multi-Purpose Keys:

- **lcc:actionId** (mandatory)
- **lcc:actionType** (mandatory)
- **lcc:actionSetId** (mandatory)
- **lcc:actionSortMatchAsNumber** (optional)
- **lcc:actionFlag** (optional)
- **lcc:actionSkip** (optional)

**lcc:actionColumnId** (mandatory)(one per ASS)

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax:  `lcc:actionColumnId [tab] [ID]`

Example:  `lcc:actionColumnId SID`

**lcc:actionColumnPos** (mandatory)(one per ASS)

The column position number, where '1' is the first column, '2' is second, etc. What column value will be used for the sort. If it is a number, see using the key 'lcc:actionSortMatchAsNumber'.

Requirement: must be a number

Syntax:  `lcc:actionColumnPos [tab] [#]`

Example:  `lcc:actionColumnPos 2`

**lcc:actionSortOrder** (optional)(one to many per ASS)

What order to sort. By default, they will be sorted in Ascending order.

**Valid values**

- Ascending
- Descending

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

Example:  `lcc:actionSortOrder Ascending`

Example Action Set: Sort

**lcc:actionId**  Sort Master On SID

Lower Columbia College
Action Keys For: Split On Character

Additional Keys that can be used from the Multi-Purpose Keys:

- lcc:actionId mandatory
- lcc:actionType mandatory
- lcc:actionSetId mandatory
- lcc:actionColumnValueMinLenToApplyRule optional
- lcc:actionColumnValueMaxLenToApplyRule optional
- lcc:actionFlag optional
- lcc:actionSkip optional

**lcc:actionColumnId** *(mandatory)* *(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax: lcc:actionColumnId [tab] [ID]

Example: lcc:actionColumnId Street

**lcc:actionColumnPos** *(mandatory)* *(one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column value to split on the character.

*Requirement: must be a number*

Syntax: lcc:actionColumnPos [tab] [#]

Example: lcc:actionColumnPos 7

**lcc:actionSplitChar** *(mandatory)* *(one per ASS)*

What character to split on.
You can also specify these white-space characters:

- [tab]: will change to a single tab character with processing
- [space]: will change to a single space character with processing

*Requirement: must be a single character*, or a White Space key word.

**Syntax:**
```
   lcc:actionSplitChar [tab] [Char]
```

**Example:**
```
   lcc:actionSplitChar ,
```

**Example:**
```
   lcc:actionSplitChar [space]
```

**lcc:actionSplitMinCols (optional)(one per ASS)**
If set, will force a minimum columns produced. If splitting the value creates less than the minimum, extra empty columns will be insert to the right until the minimum is met. For example, if your column value has "David A Mielcarek" and you split on space, that column will now contain "David" and two new columns will be inserted to the right with "A" and "Mielcarek".

*Requirement: must be a number*

**Syntax:**
```
   lcc:actionSplitMinCols [tab] [#]
```

**Example:**
```
   lcc:actionSplitMinCols 3
```

**lcc:actionSplitMaxCols (optional)(one per ASS)**
If set, will force a maximum columns produced. If splitting the value creates more than the maximum, only the quantity of columns allowed will be inserted. For example, if your column value has "David A Mielcarek" and you split on space, and you specify a maximum of '2', that column will now contain "David" and one new column will be inserted to the right with "A".

*Requirement: must be a number*

**Syntax:**
```
   lcc:actionSplitMaxCols [tab] [#]
```

**Example:**
```
   lcc:actionSplitMaxCols 3
```

**Example Action Set: Split On Character**
```
   lcc:actionId     Split Column Name, on space
   lcc:actionType    Split On Character
   lcc:actionSetId   Master
   lcc:actionSplitChar [space]
   lcc:actionSplitMinCols 3
```
lcc:actionSplitMaxCols  3
lcc:actionColumnId  Name
lcc:actionColumnPos 1
lcc:actionSkip YES

**Action Keys For: Split On Position**

Additional Keys that can be used from the Multi-Purpose Keys:

- lcc:actionId  mandatory
- lcc:actionType  mandatory
- lcc:actionSetId  mandatory
- lcc:actionColumnValue  mandatory
- lcc:actionColumnValueMinLenToApplyRule  optional
- lcc:actionColumnValueMaxLenToApplyRule  optional
- lcc:actionFlag  optional
- lcc:actionSkip  optional

**lcc:actionColumnId** *(mandatory)(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax:    lcc:actionColumnId [tab] [ID]
Example:   lcc:actionColumnId  Street

**lcc:actionColumnPos** *(mandatory)(one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. What column value to split on the character.

Requirement: must be a number

Syntax:    lcc:actionColumnPos [tab] [#]
Example:   lcc:actionColumnPos 7

**lcc:actionColumnValue** *(mandatory)(one per ASS)*

What position in the value should be split on. If the value length is smaller, a blank column will be split off.

Syntax:    lcc:actionColumnValue [tab] [#]
Example:   lcc:actionColumnValue  5
Example Action Set: Split On Position

lcc:actionId  Split Column Position, on space
lcc:actionType  Split On Position
lcc:actionSetId  Master
lcc:actionColumnId  Name
lcc:actionColumnPos 1
lcc:actionColumnValue 5
   lcc:actionSkip YES

Action Keys For: Swap Columns

Additional Keys that can be used from the Multi-Purpose Keys:
- lcc:actionId  mandatory
- lcc:actionType  mandatory
- lcc:actionSetId  mandatory
- lcc:actionColumnValueMinLenToApplyRule  optional
- lcc:actionColumnValueMaxLenToApplyRule  optional
- lcc:actionFlag  optional
- lcc:actionSkip  optional

lcc:actionColumnId  (mandatory)(one per ASS)
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.
Syntax:  lcc:actionColumnId [tab] [ID]
Example:  lcc:actionColumnId  Last Name

lcc:actionColumnPos  (mandatory)(one per ASS)
The column position number, where '1' is the first column, '2' is second, etc. What column will be swapped with another.
Requirement: must be a number
Syntax:  lcc:actionColumnPos [tab] [#]
Example:  lcc:actionColumnPos 1

lcc:actionColumn2Id  (mandatory)(one per ASS)

Lower Columbia College
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column. This column will be used to see if a supplied value exists.

Syntax: \texttt{lcc:actionColumn2Id \{tab\} [ID]}
Example: \texttt{lcc:actionColumn2Id First Name}

\textbf{lcc:actionColumn2Pos \textit{(mandatory)}\textit{(one per ASS)}}

The column position number, where '1' is the first column, '2' is second, etc. What column will be swapped with another.

\textit{Requirement: must be a number}

Syntax: \texttt{lcc:actionColumn2Pos \{tab\} [#]}
Example: \texttt{lcc:actionColumn2Pos 2}

Example Action Set: Swap Columns
\begin{verbatim}
lcc:actionId    Swap Last Name and First Name
lcc:actionType  Swap Columns
lcc:actionSetId Master
lcc:actionColumnId Last Name
lcc:actionColumnPos 1
lcc:actionColumn2Id First Name
lcc:actionColumn2Pos 2
lcc:actionSkip YES
\end{verbatim}

\textbf{Action Keys For: Trim}

Additional Keys that can be used from the Multi-Purpose Keys:
- \texttt{lcc:actionId \textit{mandatory}}
- \texttt{lcc:actionType \textit{mandatory}}
- \texttt{lcc:actionSetId \textit{mandatory}}
- \texttt{lcc:actionColumnValueMinLenToApplyRule \textit{optional}}
- \texttt{lcc:actionColumnValueMaxLenToApplyRule \textit{optional}}
- \texttt{lcc:actionFlag \textit{optional}}
- \texttt{lcc:actionSkip \textit{optional}}

\textbf{lcc:actionColumnId \textit{(mandatory)}\textit{(one per ASS)}}
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

**Syntax:** `lcc:actionColumnId [tab] [ID]

**Example:** `lcc:actionColumnId Street`

**lcc:actionColumnPos (mandatory)(one per ASS)**
The column position number, where '1' is the first column, '2' is second, etc. This value will have white-space characters removed from the left and right.

**Requirement:** must be a number

**Syntax:** `lcc:actionColumnPos [tab] [#]

**Example:** `lcc:actionColumnPos 7`

**Example Action Set: Trim**
```
lcc:actionId Trim - Street
lcc:actionType Trim
lcc:actionSetId Master
lcc:actionColumnId Street
lcc:actionColumnPos 7
lcc:actionSkip YES
```

**Action Keys For: Trim Left**

Additional Keys that can be used from the Multi-Purpose Keys:
- `lcc:actionId` mandatory
- `lcc:actionType` mandatory
- `lcc:actionSetId` mandatory
- `lcc:actionColumnValueMinLenToApplyRule` optional
- `lcc:actionColumnValueMaxLenToApplyRule` optional
- `lcc:actionFlag` optional
- `lcc:actionSkip` optional

**lcc:actionColumnId (mandatory)(one per ASS)**
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.
Syntax:
```
 lcc:actionColumnId [tab] [ID]
```
Example:
```
 lcc:actionColumnId  Street
```

**lcc:actionColumnPos (mandatory)(one per ASS)**
The column position number, where '1' is the first column, '2' is second, etc. This value will have white-space characters removed from the left.

*Requirement: must be a number*

Syntax:
```
 lcc:actionColumnPos [tab] [#]
```
Example:
```
 lcc:actionColumnPos 7
```

**Example Action Set: Trim Left**

```
 lcc:actionId  Trim Left - Street
 lcc:actionType  Trim Left
 lcc:actionSetId  Master
 lcc:actionColumnId  Street
 lcc:actionColumnPos 7
 lcc:actionSkip  YES
```

**Action Keys For: Trim Right**

Additional Keys that can be used from the Multi-Purpose Keys:
- **lcc:actionId**  mandatory
- **lcc:actionType**  mandatory
- **lcc:actionSetId**  mandatory
- **lcc:actionColumnValueMinLenToApplyRule**  optional
- **lcc:actionColumnValueMaxLenToApplyRule**  optional
- **lcc:actionFlag**  optional
- **lcc:actionSkip**  optional

**lcc:actionColumnId (mandatory)(one per ASS)**
The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

Syntax:
```
 lcc:actionColumnId [tab] [ID]
```

Lower Columbia College
Example:  lcc:actionColumnId  Street

**lcc:actionColumnPos** *(mandatory)* *(one per ASS)*

The column position number, where '1' is the first column, '2' is second, etc. This value will have white-space characters removed from the right.

*Requirement:* must be a number

*Syntax:*  lcc:actionColumnPos [tab] [#]

*Example:*  lcc:actionColumnPos 7

Example Action Set: Trim Left

lcc:actionId  Trim Right - Street
lcc:actionType  Trim Right
lcc:actionSetId  Master
lcc:actionColumnId  Street
lcc:actionColumnPos 7
  lcc:actionSkip  YES

**Action Keys For: Upper Case**

Additional Keys that can be used from the Multi-Purpose Keys:

- lcc:actionId  *mandatory*
- lcc:actionType  *mandatory*
- lcc:actionSetId  *mandatory*
- lcc:actionColumnValueMinLenToApplyRule  *optional*
- lcc:actionColumnValueMaxLenToApplyRule  *optional*
- lcc:actionFlag  *optional*
- lcc:actionSkip  *optional*

**lcc:actionColumnId** *(mandatory)* *(one per ASS)*

The ID used for a column. This is only used when viewing logs/reports to make it easier to know which column position is being affected. Should describe the values in the column.

*Syntax:*  lcc:actionColumnId [tab] [ID]

*Example:*  lcc:actionColumnId  Street
**lcc:actionColumnPos** *(mandatory)*(one per ASS)

The column position number, where '1' is the first column, '2' is second, etc. This value will be upper cased.

**Requirement:** must be a number

**Syntax:**  
\[lcc\text{-}actionColumnPos\] 

**Example:**  
\[lcc\text{-}actionColumnPos\text{7}\]

---

**Logic File – Category: Target Columns**

The Target Columns category instructs the program what column from the Master to place in what 'Target' file.

Each Target Column is defined by creating a Target Column Settings Set (TCSS).

Each TCSS is a File defined using Keys starting with: **lcc:targetColumn**

Each TCSS starts with the Key: **lcc:targetColumnId**

**lcc:targetColumnId** *(mandatory)*(one per FSS)

The ID used for specifying what the Target Column is. Each File Id must be unique, but, can be any value. Should describe what the file contains.

**Syntax:**  
\[lcc\text{-}targetColumnId\] 

**Example:**  
\[lcc\text{-}targetColumnId\text{First Name}\]

**lcc:targetColumnSourcePos** *(mandatory)*(one per ASS)

The column position number, where '1' is the first column, '2' is second, etc. What column from the Master will be used.

**Requirement:** must be a number
Syntax: \texttt{lcc:targetColumnSourcePos} [\texttt{tab}] [#]
Example: \texttt{lcc:targetColumnSourcePos 1}

\textbf{\texttt{lcc:targetColumnTargetPos}} \textit{(mandatory)} \textit{(one per ASS)}
The column position number, where '1' is the first column, '2' is second, etc. What column in the Target File will this value be placed into.
\textit{Requirement: must be a number}
Syntax: \texttt{lcc:targetColumnTargetPos} [\texttt{tab}] [#]
Example: \texttt{lcc:targetColumnTargetPos 1}

\textbf{\texttt{lcc:targetColumnTarget}} \textit{(mandatory)} \textit{(one per ASS)}
What Target File this value will be written to.
\textit{Requirement: must be a number}
Syntax: \texttt{lcc:targetColumnTarget} [\texttt{tab}] [...]
Example: \texttt{lcc:targetColumnTarget Academic Works Import}

\section*{Definitions}
\textbf{Camel Case} - changing the first character of each word to a capital letter, all other letters set to lower case
\textbf{Data Set} - a delimited file loaded into memory
\textbf{File Settings Set} - (FSS) a set of settings for a file
\textbf{Lower Case} - changing all letters to non-caps
\textbf{Record} - a row of information (single line from file/memory)
\textbf{Upper Case} - changing all letters to caps
\textbf{White Space Character} - a character this is invisible on screen. Like a [space], [tab], etc.

\section*{Modifications}

\begin{tabular}{|l|l|l|}
\hline
\textbf{NAME} & \textbf{DATE} & \textbf{MODIFICATION} \\
\hline
David Mielcarek & 1/11/2016 & Created \\
David Mielcarek & 1/12/2016 & Added more examples, fixed some erratas \\
\hline
\end{tabular}
<table>
<thead>
<tr>
<th>David Mielcarek</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1/15/2016</td>
<td>Added Numbers And A Single Period to New Data Format options</td>
</tr>
<tr>
<td></td>
<td>1/19/2016</td>
<td>Added new actions</td>
</tr>
<tr>
<td></td>
<td>1/21/2016</td>
<td>Added new actions</td>
</tr>
<tr>
<td></td>
<td>2/1/2016</td>
<td>Change Record Header key format. Added Unix/Microsoft Time Stamp to Convert To Key options. Added Action Flags</td>
</tr>
<tr>
<td></td>
<td>3/14/2016</td>
<td>Added new format 'YYYY-MM-DD' for lcc:actionColumnNewFormat</td>
</tr>
<tr>
<td></td>
<td>3/15/2015</td>
<td>Added Keep First, Keep Last to Combine Duplicates, Replace</td>
</tr>
<tr>
<td></td>
<td>3/16/2016</td>
<td>Special Value replacing in Column Value</td>
</tr>
<tr>
<td></td>
<td>3/21/2016</td>
<td>Added lcc:actionFlag value 'DecimalPlaces: #'</td>
</tr>
<tr>
<td></td>
<td>3/23/2016</td>
<td>Added Action Type 'Insert Line At Each Record Set', new Convert To Original Format 'HHMMAM/PM', New Format 'HH:MM AM/PM'</td>
</tr>
<tr>
<td></td>
<td>3/28/2016</td>
<td>Added lcc:fileRecordsSkipLoading</td>
</tr>
<tr>
<td></td>
<td>5/9/2016</td>
<td>Added lcc:ActionFlag values 'Right To Left', 'Keep Left Over'</td>
</tr>
<tr>
<td></td>
<td>5/10/2016</td>
<td>Added Action Type &quot;Split On Position&quot;</td>
</tr>
<tr>
<td></td>
<td>6/16/2016</td>
<td>Added 'Current' to key lcc:actionTwoToFourYearThreshold</td>
</tr>
<tr>
<td></td>
<td>2/24/2017</td>
<td>Fixed spelling mistake in manual for key 'lcc:actionSortMatchAsNumber'</td>
</tr>
</tbody>
</table>

End of document