

# IccTextReformatter Manual

## Contents

Description .....	1
lccTextReformatter .....	1
Installation .....	2
Using.....	2
Types Of Reformatting .....	3
Flags .....	5
Logic File Description/Syntax .....	6
Logic File .....	6
Logic File - Examples.....	7
Definitions.....	7
Modifications .....	8

---

## Description

This document describes how to use the lccTextReformatter.

### **lccTextReformatter**

is a Windows program created to reformatting of text.

Some examples reformatting options are:

- make PLSQL/TSQL queries more readable
- translate between PLSQL to TSQL

- transate between TSQL to PLSQL
- convert MAC address to different format standards, i.e. #####, ##:##:##:##:##, etc.
- turn lines into SQL Values, i.e. 'a','b','c'

Also supports:

- indenting
- camel case
- lower casing
- upper casing

---

## Installation

- copy the lccTextReformatter.exe to a folder
- (optiona) create/edit the Logic File as desired
- run the lccTextReformatter.exe

By default, the program will look for a Logic File "lccTextReformatter-logic.txt".

Optional: you can provide Logic Files by running the program with additional parameters:

**Syntax:** `lccTextReformatter.exe lcc:logicPath [...Logic File...]`

**Example:** `lccTextReformatter.exe lcc:logicPath lccTextReformatter-logic.txt`

---

## Using

- Run the program
- Choose the 'Type' of reformatting (see section "Types Of Reformatting")
- Choose the 'Flags' (see section "Flags")
- Click tab "Source"
  - Either:
    - Click: Load From File, to load text from a file
    - or-
    - Click: Paste, to paste text from your clipboard
    - or-

- Manually paste text into the box
- Click tab "Results"
  - Click: Process, this will reformat the Source text and place in the Results

**Optional steps:**

- Click tab "Source"
  - Click "Copy", to copy the Source text to your clipboard
  - Click "Save", to save the Source text to a file
- Click tab "Results"
  - Click "Copy", to copy the Results text to your clipboard
  - Click "Save", to save the Results text to a file

---

## Types Of Reformatting

### Flags Only

- only reformats according to the Flags chosen, not Type used.

### Lines To Comma Delimited Values

- Converts lines of text into comma delimited Values.

- 

For example, if you provided:

one  
two  
three

The output would be:

one,two,three

### Lines To SQL Values

- Converts lines of text into SQL Values. This is handy when you want to use a list of lines (values) in a SQL statement.

For example, if you provided:

one  
two

three

The output would be:

'one','two','three'

#### Lines To TAB Delimited Values

- Converts lines of text into TAB delimited Values.
- 

For example, if you provided:

one

two

three

The output would be:

one    two    three

#### MAC To ...

- Changes MAC addresses to different formats

#### New Line At < And >

- Places a new line before each '<' and after each '>', making HTML/XML/etc. (those using '<' and '>' as structure) more readable.

#### PLSQL Readable

- Processes Flags (see section "Flags")
- Places line breaks at specific commands, like "Create", "While", commas, etc.
- Changes all double spaces into single spaces

#### PLSQL To SQL

- Changes all double quotes '"' into left '[' and right ']' brackets
- Changes all double pipes '||' into a plus sign '+'

### **TSQL Readable**

- Processes Flags (see section "Flags")
- Places line breaks at specific commands, like "Create", "While", commas, etc.
- Changes all double spaces into single spaces

### **TSQL To PLSQL**

- Changes all left '[' and right ']' brackets into double quotes ''''
- Changes all plus signs '+' into double pipes '||'

---

## **Flags**

### **Camel Case**

Changes the first character to an upper case, and all others to lower for each word.

### **CRLF To LF**

Changed all Carriage Return/Line Feed characters into Line Feed (i.e. from Microsoft standard to Linux)

### **Indent**

Indents at specific line breaks.

### **LF To CRLF**

Changed all Line Feed characters into Carriage Return/Line Feed (i.e. from Linxu standard to Microsoft)

### **Lower Cased**

Changes all letters into lowercase.

## Upper Cased

Changes all letters into uppercase.

---

## Logic File Description/Syntax

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

The Logic File uses the syntax.

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

**Example:** **lcc:key**     **value**

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

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

---

## Logic File

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

Auto process the source/results as soon as the program opens.

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

**Example:**     **lcc:autoProcess**     **YES**

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

Auto load a file into Source.

**Syntax:**        **lcc:sourceLoadFromFilePath** *[tab]* **[...path...]**

**Example:**     **lcc:sourceLoadFromFilePath**        **\\server\share\$\folder\ourquery.txt**

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

Auto load a file into Source.

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

Example: `lcc:sourceLoadFromFilePath \\server\share$\folder\ourquery.txt`

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

Auto set the Results Type.

The value can be any of the Results Types listed in the drop down, ex:

- TSQL
- PLSQL To SQL

Syntax: `lcc:resultsType [tab] [...type...]`

Example: `lcc:resultsType PLSQL To SQL`

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

Auto check a results Flag.

The value can be any of the results Flags listed:

- Indent
- Upper Cased

Syntax: `lcc:resultsFormatFlag [tab] [...type...]`

Example: `lcc:resultsFormatFlag Indent`

---

## Logic File - Examples

```
lcc:autoProcess YES
lcc:sourceLoadFromFilePath dataLinkPLSQLQuery.txt
lcc:resultsType PLSQL To TSQL
    lcc:resultsFormatFlag Indent
    lcc:resultsFormatFlag Upper Cased
```

---

## Definitions

**Column** - see Field.

**PLSQL** - Oracel/PeopleSoft SQL

**SQL** - Structured Query Language

**TSQL** - Microsoft Transactional SQL

---

## Modifications

<b>NAME</b>	<b>DATE</b>	<b>MODIFICATION</b>
David Mielcarek	20200228	Created
David Mielcarek	20200403	Added MAC To ...
David Mielcarek	20200512	Added Lines To SQL Values, Camel Case
David Mielcarek	20210115	Added Flags "CRLF To LF" and "LF To CRLF"
David Mielcarek	20210401	Added Lines To Comma/TAB Delimited

---

End of document