

# lccReplaceCharInFile Manual

## Contents

Description.....	1
Installation.....	1
Running The Program.....	2
Logic File.....	2
Purpose.....	2
Syntax .....	2
Keys.....	2
White Space Byte Names .....	4
Logic File Examples .....	5
Command Line Examples .....	5
Definitions .....	5
Modifications.....	6

---

## Description

This document describes the lccReplaceCharInFile program.

The lccReplaceCharInFile is a Command Line program that replaces bytes (char) from a file.

The program is set by a Logic File and/or parameters supplied in the command line.

Process Topology:

- looks for a default Logic File (lccReplaceCharInFile-logic.txt) and uses if found
- open desired file
- create target file
- read bytes
- if match found, replace
- write bytes to target file

---

## Installation

- copy lccReplaceCharInFile.exe to a folder
- (optional) build a Logic File

(run)

---

## Running The Program

Run from the command line, using the following command.

### Syntaxes

#1: `lccReplaceCharInFile.exe`

Looks for the a default Logic File and uses those settings.

#2: `lccReplaceCharInFile.exe lcc:logicPath (Logic File)`

Looks for a non-default Logic File and uses those settings.

#3: `lccReplaceCharInFile.exe [...parameters, aka Key/Values...]`

Looks for the a default Logic File and uses command line parameters for settings.

### Examples

#1 `lccRecordsVerifier.exe`

### Examples

#2 `lccRecordsVerifier.exe lcc:logicPath ourLogic.txt`

### Examples

#3 `lccRecordsVerifier.exe lcc:sourceFile file1.txt lcc:targetFile file1-changed.txt lcc:replace "[CR]" lcc:with "[tab]" lcc:replace "[LF]" lcc:with "[skip]"`

---

## Logic File

### Purpose

The Logic File instructs the program on how to send the Request and process the Response.

### Syntax

The Logic File expects settings to be supplied in the following manner. However, any lines that do not match will be ignore, i.e. can be used to place comments/etc.

### Syntax

`[key] [tab] [value]`

### Example

`lcc:exampleKey      exampleValue`

### Keys

**lcc:sourceFile** *(mandatory)(one per Logic File)*

The source file to read from.

### Syntax

`lcc:sourceFile [tab] [value]`

**Example**

`lcc:sourceFile file1.txt`

**lcc:targetFile** *(mandatory)(one per Logic File)*

The target file to write to.

**Syntax**

`lcc:targetFile [tab] [value]`

**Example**

`lcc:targetFile file1-changed.txt`

**lcc:startAfterPos** *(optional)(one per Logic File)*

If supplied, instructs the program to not compare bytes for replacement until after this file position. For example, if you supply '5', the first 5 bytes will be written to the target file, with no comparing.

**Syntax**

`lcc:startAfterPos [tab] [number]`

**Example**

`lcc:startAfterPos 5`

**lcc:heartBeat** *(optional)(one per Logic File)*

If supplied, will display a Heart Beat every # bytes read.

**Syntax**

`lcc:heartBeat [tab] [number]`

**Example**

`lcc:heartBeat 1000000`

**lcc:replace** *(mandatory)(one to many per Logic File)*

What byte(s) to replace. Each of these keys will be paired with the next mandatory '**lcc:with**' key. Certain 'names' can be provided to specify white-space bytes. See the White Space Byte Names section.

More than one byte can be supplied. Separate each with a [tab] character.

A number can be supplied for the byte using the syntax:

`lcc:number=[number]`

**Syntax**

`lcc:replace [tab] [byte]`

**Example #1** *(a single byte)*

`lcc:replace X`

**Example #2** (a white-space name)

lcc:replace [CR]

**Example #3** (a number)

lcc:replace lcc:number=123

**Example #4** (multiple bytes, spelling out 'dog' in unicode)

lcc:replace lcc:number=0d lcc:number=0 o lcc:number=0 g

**lcc:with** (mandatory)(one to many per Logic File)

What byte to use when replacing. Each of these keys will be paired with the previous mandatory 'lcc:replace' key. Certain 'names' can be provided to specify white-space bytes. See the White Space Byte Names section.

You can also supply a value of '[skip]' that will instruct the program to skip the byte found with the paired 'lcc:replace' key. i.e. that byte will not be written to the target file.

More than one byte can be supplied. Separate each with a [tab] character.

A number can be supplied for the byte using the syntax:

lcc:number=[number]

**Syntax**

lcc:with [tab] [byte]

**Example #1** (a single byte)

lcc:with X

**Example #2** (a white-space name)

lcc:with [CR]

**Example #3** (a number)

lcc:with lcc:number=123

**Example #4** (multiple bytes, spelling out 'cat' in unicode)

lcc:with lcc:number=0c lcc:number=0 a lcc:number=0 t

---

## White Space Byte Names

The following names can be used with the 'lcc:replace' and 'lcc:with' key/values to specify white-space bytes. This also allows editing/reading parameters/Logic File values easier.

NAME	DEFINITION	ASCII CODE
[CR]	Carriage Return	13
[LF]	Line Feed	10
[tab]	Tab	9
[space]	Space	32

---

## Logic File Examples

```
=====
CREATED
David Mielcarek, Aug 25, 2015

DEFINTION
Replaces the following bytes:
    All Carriage Returns with a space.
    All Line Feeds are removed.
    All 'e's become '@'.
=====
lcc:sourceFile 3.txt
lcc:targetFile 3-changed.txt

lcc:replace [CR]
lcc:with [space]

lcc:replace [LF]
lcc:with [skip]

lcc:replace e
lcc:with @

lcc:replace lcc:number=13 lcc:number=10 d o g
lcc:with lcc:number=9c a t
-----
```

---

## Command Line Examples

Looks for the a default Logic File and uses those settings.

`lccRecordsVerifier.exe`

Looks for a non-default Logic File and uses those settings.

`lccRecordsVerifier.exe lcc:logicPath ourLogic.txt`

Looks for the a default Logic File and uses command line parameters for settings.

`lccRecordsVerifier.exe lcc:sourceFile file1.txt lcc:targetFile file1-changed.txt lcc:replace "[CR]" lcc:with "[tab]" lcc:replace "[LF]" lcc:with "[skip]"`

---

## Definitions

**ASCII** - American Standard Code for Information Interchange

ref: <https://en.wikipedia.org/wiki/ASCII>

---

## Modifications

<b>NAME</b>	<b>DATE</b>	<b>MODIFICATION</b>
David Mielcarek	8/25/2015	Created
David Mielcarek	9/8/2015	Added multiple byte option for replace and with keys. Added new keys lcc:startAfterPos, lcc:heartBeat

---

End of document