

# lccSQLAdhocStatements

## White Paper

### Contents

Description .....	1
How To Run Examples .....	1
Logic File Description/Syntax .....	2
Keys .....	2
Logic File Example .....	5
Definitions .....	6
Modifications .....	6

---

### Description

This document describes how to use the *lccSQLAdhocStatements* program.

The lccSQLAdhocStatements program is a console application that creates SQL adhoc statement(s) from lines/columns read from a delimited file(s).

---

### How To Run Examples

Providing settings from a Logic File

```
lccWakeOnLAN.exe lcc:logicFile lccWakeOnLAN-logic.txt
```

---

## Logic File Description/Syntax

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**

---

## Keys

If using as a command line parameter, use a space between a key and value.

If using a Logic File, use a tab between a key and value.

See section Flag Replace for value that can be modified on-the-fly.

**lcc:logPath** (optional)

*[ one per run ]*

What root filename to use on logs. The date "-YYYYMMDD" and ".log" will auto appended to the name.

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

Example: **lcc:logPath**            **lccSQLAdhocStatements-log**

**lcc:method** (optional)

*[ one per run ]*

Tells the program what to do.

- **read and create** : read source file(s) and create target file(s)

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

Example: **lcc:method broadcast**

**lcc:flag** (optional)

*[ one per to many run ]*

Flags to control the program.

- **skip header rows:** the number of header rows to skip from the source file. Must be a number.

Syntax: **lcc:flag** [tab] [...flag...] [tab] [...value...]

Example: **lcc:flag skip header rows 1**

**lcc:sourceSetId** (mandatory)

*[ one to many per run, one per set ]*

Create a new Source Set. A source set defines the source file, delimiter, columns, SQL statements, target file, etc.

The 'id' can be anything and is only used in logging.

Syntax: **lcc:sourceSetId** [tab] [...id...]

Example: **lcc:sourceSetId ourId**

**lcc:sourceSetDelimiter** (mandatory)

*[ one to many per run, one per set ]*

What delimiter is used in the source file.

Special white-space characters can be defined as:

- **[SPACE]:** a space character
- **[TAB]:** a tab character

Syntax: **lcc:sourceSetDelimiter** [tab] [...delimiter...]

Example: **lcc:sourceSetDelimiter ,**

Example #2: **lcc:sourceSetDelimiter [TAB]**

**lcc:sourceSetColumns** (mandatory)

*[ one to many per run, one per set ]*

What columns (comma delimited) are in the source file. Any lines that do not define at least that many columns will be skipped.

**Syntax:** `lcc:sourceSetColumns [tab] [...column name,column name...]`

**Example:** `lcc:sourceSetColumns col1,col2,col3`

**lcc:sourceSetPath** (mandatory)

*[ one to many per run, one per set ]*

Path to the source file.

**Syntax:** `lcc:sourceSetPath [tab] [...path...]`

**Example:** `lcc:sourceSetPath \\server\share$\directory\ourFile.csv`

**lcc:sourceSetSQLStatement** (mandatory)

*[ one to many per run, one to many per set ]*

A SQL statement(s) to write to the target path.

The following flags will be replaced in each line:

- **[lccFlag:columns]:** all columns, separated with commas
- **[lccFlag:columnValues]:** all column values, enclosed in single quotes and separated by commas
- **[lccFlag:column:...]:** a column name
- **[lccFlag:columnValue:...]:** a column value
- 

**Syntax:** `lcc:sourceSetSQLStatement [tab] [...SQL statement...]`

**Example:** `lcc:sourceSetSQLStatement INSERT INTO dbo.ourTable ([lccFlag:columns]) VALUES ([lccFlag:columnValues]);`

**Example #2 (multiple):**

`lcc:sourceSetSQLStatement INSERT INTO dbo.ourTable ([lccFlag:columns]) VALUES ([lccFlag:columnValues]);`

```
UPDATE dbo.ourTable2 SET  
[lccFlag:column:DESCR]='[lccFlag:columnValue:DESCR]',TERM_BEGIN_DT='[lccFlag:columnValue:TERM_B  
EGIN_DT]' WHERE ourColumns='ourValue';
```

**lcc:sourceSetTargetPath** (mandatory)

*[ one to many per run, one per set ]*

Path to the target file.

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

**Example:** **lcc:sourceSetTargetPath** \\server\share\$\directory\ourCommands.sql

---

## Logic File Example

Loads two files and creates two files. The first file creates a single SQL statement per source line. The second file creates two SQL statements.

```
lcc:logPath lccSQLAdhocStatements-log
```

```
lcc:method read and create
```

```
lcc:method skip header rows 1
```

```
lcc:sourceSetId PS_SESSION_TBL
```

```
lcc:sourceSetDelimiter ,
```

```
lcc:sourceSetColumns INSTITUTION,ACAD_CAREER,STRM,SESSION_CODE,SESS_BEGIN_DT,SESS_END_DT,ENROLL_OPEN_DT
```

```
lcc:sourceSetPath \\server\share$\directory\ourFile.csv
```

```
lcc:sourceSetSQLStatement INSERT INTO dbo.ourTable ([lccFlag:columns]) VALUES ([lccFlag:columnValues]);
```

```
lcc:sourceSetTargetPath \\server\share$\directory\PS_SESSION_TBL.sql
```

```
lcc:sourceSetId PS_TERM_TBL
```

```
lcc:sourceSetDelimiter ,
```

```
lcc:sourceSetColumns
```

```
INSTITUTION,ACAD_CAREER,STRM,DESCR,DESCRSHORT,TERM_BEGIN_DT,TERM_END_DT,SESSION_CODE
```

```
lcc:sourceSetPath \\server\share$\directory\ourFile.csv
lcc:sourceSetSQLStatement      INSERT INTO dbo.ourTable2 ([lccFlag:columns]) VALUES
([lccFlag:columnValues]);
lcc:sourceSetSQLStatement      UPDATE dbo.ourTable2 SET
[lccFlag:column:DESCR]='[lccFlag:columnValue:DESCR]',TERM_BEGIN_DT='[lccFlag:columnValue:TERM_BEGIN_DT]'
WHERE ourColumns='ourValue';
lcc:sourceSetTargetPath \\server\share$\directory\PS_TERM_TBL.sql
```

---

## Definitions

---

## Modifications

NAME	DATE	MODIFICATION
David Mielcarek	20230724	Created

---

End of document