Laser-Scan Ltd.

LITES2

LCMSQUASH User Guide

Science Park, Milton Road, Cambridge, England CB4 4FY tel: +44 (0) 1223 420414

Document "LITES2 - LCMSQUASH User Guide" Category "USER"

Document Issue 1.0	Sunil Gupta	2-Jul-1992
Document Issue 1.1	Clarke Brunt	20-Oct-1992
Document Issue 1.2	Paul Hardy	3-Feb-1993
Document Issue 1.3	Clarke Brunt	3-Jun-1994

1 **FUNCTION**

This document describes the LCMSQUASH utility. It is to be read as a supplement to the LITES2 Reference Manual and the LITES2 User's Guide.

LCMSQUASH is a LITES2 utility to preprocess LCM (LITES command) files. This reduces the time taken for LITES2 to load the files, and can also provide a distinction between the 'source code' (the unprocessed file, which is easily readable) and the 'executable code' (the processed file, which is much less easily read by humans).

LCMSQUASH provides the following functions:

- removal of all comments from LCM files.
- truncation of LITES2 primary commands to 4 characters.
- reduction of multiple-white space.
- compression of multiple commands to fit onto lines of 255 characters or less.
- optional encryption of the output LCM file.
- optional decompression of LCM files to one command per line.
- production of a listing file to enable debugging.
- simple syntax checking.
- summary of all macro and variable names declared and used.
- diagnostic warnings.

2 **FORMAT**

```
$ LCMSQUASH input-file-spec  [output-file-spec]
```

Command qualifiers	**Defaults**
/[NO]CHECK=QUOTE	/NOCHECK
/[NO]CODE=integer	/NOCODE
/[NO]COMMENT=string	/NOCOMMENT
/[NO]EXPLODE=file	/NOEXPLODE
/[NO]KEY=string	/NOKEY
/[NO]LOG	/NOLOG
/[NO]LIST=file	/NOLIST
/[NO]QUIET	/NOQUIET

3 **PROMPT**

```
_Input_file:            input-file-spec
_Output_file:           output-file-spec
```

4 **PARAMETERS**

Input-file-spec

- specifies the LCM file which is processed. Any part of the file specification which is not supplied will be taken from the default specification 'LSL$LITES2CMD:.LCM'. The input file is required.

Output-file-spec

- specifies the LCM file which is to be output. The default file name is the name part of the input file, with the extension '.LCM'. By default, the output file is written to the current directory. LCMSQUASH attempts to prevent the creation of an output file which is just a higher version of the input file, since it is usually important that the original file is kept intact. An error will be generated under these circumstances, unless explicit version numbers are included on the file names.

5 **COMMAND QUALIFIERS**

/CHECK=QUOTE
/NOCHECK (default)

- outputs warning messages if a substituted variable does not have a quote after it. This check is not performed by default, since LITES2 accepts variables without the closing quote.

/CODE=integer
/NOCODE (default)

- specifies an encryption code, which is a small integer allocated to users by Laser-Scan. The code is written into the first line of the encrypted output file and identifies the particular encryption to LITES2.

This qualifier must be used in conjunction with the /KEY qualifier.

/COMMENT=string
/NOCOMMENT (default)

- specifies a comment string which is written into the first line of the output LCM file following a "!". It may be used, for example, to include a copyright notice in the output file. If the string contains lower case letters, or spaces, then it must be enclosed in double quotation marks. Within double quotes, use two double quotes to represent a single one.

/EXPLODE=filename
/NOEXPLODE (default)

- generates a file with commands taken from the input LCM file, such that only one command is on each line.

The default file name is the name part of the input file, with the extension '.EXP'. By default, the explode file is written to the current directory. The normal output file is still produced in addition to the explode file. If this is not required, then the null device (NL:) may be specified as the output.

/KEY=string
/NOKEY (default)

- specifies an encryption key, which is a string chosen by application developers to go with the encryption code allocated to them by Laser-Scan. Once chosen, and communicated to Laser-Scan, the same key must always be used with a particular code. The key string can contain up to 20 characters in the ASCII range 32-126 (the printable characters). If it contains lower case letters, or spaces, then it must be enclosed in double quotation marks. Within double quotes, use two double quotes to represent a single one.

This qualifier must be used in conjunction with the /CODE qualifier.

/LIST=filename
/NOLIST (default)

- generates a listing file containing numbered source lines with LCMSQUASH warning messages placed in context.

  The default file name is the name part of the input file, with the extension '.LIS'. By default, the listing file is written to the current directory.

/LOG
/NOLOG (default)

- causes the output from LCMSQUASH to include the input source lines as well as any warning messages.

/QUIET=filename
/NOQUIET (default)

- suppresses warning messages from being output.

6 **DESCRIPTION**

LCMSQUASH is an LCM file pre-processing utility.

The following pre-processing operations are used:

- All commands which are preceded by a percent symbol % are assumed to be valid LITES2 primary commands and are truncated to 4 characters in length. These are guaranteed to be unique within LITES2. Commands not preceded by a "%" are assumed to be macro calls and are not truncated.

- The output LCM file contains no comments. Any text after "!" on a line is removed. For this reason, it is usually desirable to keep the original LCM files, in addition to the output files from LCMSQUASH.

- All lines containing only white-space (spaces and TABs) are removed. White-space at the start or end of a command is removed.

- The truncation, removal of white-space and comments reduces the length of LITES2 command constructs. Multiple commands are concatenated with a hash "#" separator so that the resultant line length is less than 256 characters.

- LCMSQUASH can optionally encrypt LCM files, so that end users cannot trivially understand or modify them. This facility is accessed using the /CODE and /KEY qualifiers. Users developing LCM applications who wish to take advantage of this facility should apply to Laser-Scan to be allocated one or more encryption codes for their applications. These are small integers, a register of which will be maintained by Laser-Scan. For each code, the developer should choose a unique encryption key, which is a string of up to 20 characters (spaces are allowed). The chosen keys must be communicated to Laser-Scan. In order for a user to use the encrypted application, a special LITES2 licence file must be issued to them by Laser-Scan, which allows LITES2 to decrypt the files.

  Encrypted LCM files begin with the characters "#%#", followed by the encryption code. The rest of the first line may consist of a comment, preceded by "!". LCMSQUASH can include such a comment by the use of the /COMMENT qualifier. The comment part of the first line may be changed after encryption, but any changes whatsoever to the encrypted lines following will prevent decryption of the file by LITES2.

- LCMSQUASH keeps an internal list of all variable and macro names that it encounters. These are used to generate warnings when names are used before declaration. A summary of the names that were used and those that were declared is output at the end of processing. Note that LCM files may reference other LCM files using the @ command. Names that were not defined in input LCM file may well have been declared in other referenced LCM files. LCMSQUASH does not currently process referenced files. This feature may be added at a later date.

  LCMSQUASH will only recognise the declaration of a variable or macro if the DECLARE or MACRO command is preceded by the "%" escape character.

- LCMSQUASH checks for leading and trailing quotes in variable substitutions. The warning message about the trailing quote being missing is suppressed by default, since this is still valid LITES2 syntax. LCMSQUASH does not know whether substitution is currently enabled, nor whether a variable is an array or not. This could lead to occasional spurious warnings. The output file will not however be affected.

- The /EXPLODE qualifier may be used to expand an LCM file (possibly one already squashed) so that each command is on a separate line. This essentially converts hash-separators in the input file to <CR>s in the explode file, which allows squashed files to be presented in a more readable form. This cannot be used with files which are already encrypted.

- The summary typed at the end of the run displays the variable and macro names which were declared in the input LCM file, and also any that were used without being declared. Note that system variables which are preceded by a dollar "$" character are not included in these lists.

- All warning that are generated by LCMSQUASH are diagnostic. It is up to the user to decide whether to take and action. This non-corrective approach ensure that the context and structure of the input LCM data is maintained in the output.

7 **EXAMPLES**

In the following LCMSQUASH processing example, LCMSQUASH is being used in its default mode to `squash' and report possibly useful information about a file containing a substantial LITES2 macro.

```
$ LCMSQUASH HERE:XGIS_AP_HAR_ANNO HERE:XGIS_AP_HAR_ANNO_SQUASHED
%LCM-W-WARN, Name not declared: UT_SCREEN_HEIGHT
%LCM-W-WARN, Name not declared: I
%LCM-W-WARN, Name not declared: STRING
%LCM-W-WARN, Name not declared: _REAL
%LCM-W-WARN, Name not declared: W
%LCM-W-WARN, Name not declared: Z
%LCM-W-WARN, Name not declared: _INT
%LSLLIB-W-EOF, end of file
%LCM-I-TXT,
%LCM-I-TXT, ------------------------------------------------------------------
%LCM-I-TXT,                                DIAGNOSTICS
%LCM-I-TXT, ------------------------------------------------------------------
%LCM-I-TXT, The following variable or macro names were not declared.
%LCM-I-TXT, Some may be primary commands used without a %.
%LCM-I-TXT, If the some of the names are present in the lists following this
%LCM-I-TXT,  one, then the names were used prior to declaration.
                        I
                        STRING
                        UT_SCREEN_HEIGHT
                        W
                        Z
                        _INT
                        _REAL
%LCM-I-TXT, ------------------------------------------------------------------
%LCM-I-TXT, The following macro names were defined:
                        _HA_ANNOTATION
%LCM-I-TXT, ------------------------------------------------------------------
%LCM-I-TXT, The following variable names were declared:
                        HA_FH
                        HA_FX
                        HA_FY
                        HA_H
                        HA_I
                        HA_J
                        HA_JMINX
                        HA_JMINY
                        HA_L
                        HA_LFC
                        HA_MAXX
                        HA_MAXY
                        HA_MINX
                        HA_MINY
                        HA_TX
                        HA_TY
                        HA_X
                        HA_Y
%LCM-I-TXT, ------------------------------------------------------------------
 ELAPSED:    0 00:00:00.84  CPU: 0:00:00.27  BUFIO: 7  DIRIO: 9  FAULTS: 176
```

7.1 **MESSAGES (SUCCESS)**

These messages are used to indicate that the program has succeeded in performing some action, and do not require any user action.

NORMAL, LCMSQUASH function return was TRUE (ie success)

Explanation: This message is used internally by LCMSQUASH to indicate whether a subroutine has completed successfully or not.

User action: None.

7.2 **MESSAGES (INFORMATIONAL)**

These messages give information only, and require no immediate action by the user. They are used to provide information on the current state of the program, or to supply explanatory information in support of a warning or error message.

INFO, %S

Explanation: A diagnostic message useful for debugging purposes

User action: None

TXT, %S

Explanation: Miscellaneous operating messages

User action: None

7.3 **MESSAGES (WARNING)**

These messages are output when an error has occurred that can be corrected immediately by the user or that the program will attempt to overcome.

WARN, %S

Explanation: A general purpose warning message

User action: None

7.4 **MESSAGES (ERROR)**

These messages indicate an error in processing which will cause the program to terminate. The most likely causes are a corrupt or otherwise invalid input file, or an error related to command line processing and file manipulation.

ABORT, LCMSQUASH function return was FALSE (ie error)

Explanation: This message is used internally by LCMSQUASH to indicate whether a subroutine has completed successfully or not.

User action: None.

FILROPEN, error opening file "%S" for input

Explanation: Some form of error occurred in opening one of the input files. The program will exit.

User action: Depends upon the associated LSLLIB messages.

FILWOPEN, error opening file "%S" for output

Explanation: Some form of error occurred in opening one output files. The program will exit.

User action: Depends upon the associated LSLLIB messages.

ILLEGALKEY, illegal key string

Explanation: The string given after the /KEY qualifier contained an illegal character, or was too long. The program will exit.

User action: Specify a key which only contains characters in the ASCII range 32-126, and which is not too long.

7.5 **MESSAGES (OTHER)**

In addition to the above messages which are generated by the program itself, other messages may be produced by the command line interpreter (CLI) and by Laser-Scan libraries. In particular, messages may be generated by the IFF library and by the Laser-Scan I/O library, LSLLIB. IFF library messages are introduced by '%IFF' and are documented in the IFF library users' guide. In most cases IFF errors will be due to a corrupt input file, and this should be the first area of investigation. If the cause of the error cannot be traced by the user, and Laser-Scan are consulted, then the output file should be preserved to facilitate diagnosis. LSLLIB messages are introduced by '%LSLLIB' and are generally self-explanatory. They are used to explain the details of program generated errors.