

**VAX Language-Sensitive Editor (LSEDIT)
Quick Reference Guide, version 2.0,
for use with the
L^AT_EX Environment**

April 12, 1988

This booklet provides a brief summary of the commands and functions available with LSEDIT and the L^AT_EX environment, corresponding to L^AT_EX version 2.09.

By Kent McPherson
SLI Avionic Systems Corp.
Grand Rapids Operation

Contents

1	Introduction	1
2	LSEDIT Definitions	1
3	LSEDIT Commands	1
4	Conventions Within The \LaTeX Environment	1
5	Invoking LSEDIT	2
6	Supported \LaTeX Document Styles	2
7	Running \LaTeX From Within LSEDIT	3
8	Accessing Help for Placeholders and Tokens	4
A	Example Page Layout	6
B	Example Type Styles	8
C	Example Type Sizes	9

Scope and Intent of this Guide

This guide presents information about the Language-Sensitive Editor (LSEDIT) and the environment defined for L^AT_EX. It is a supplement to the *VAX Language-Sensitive Editor User's Guide* and *L^AT_EX, A Document Preparation System, User's Guide & Reference Manual*.

This guide acts as a summary and memory refresher for the commands and functions covered in the manuals mentioned above. It is *not* intended to replace either of these manuals. It should supplement these documents and provide information that is specific to this environment.

The information in this document is subject to change without notice.

Conventions

Various symbols and syntax conventions are used throughout this guide.

Symbol	Meaning
\$	System prompt
RETURN	Return key
CTRL/X	Control key + main keyboard key
GOLD	PF1 key
HELP	PF2 key
KP 0 ...	Numeric keypad key

1 Introduction

The Language-Sensitive Editor, hereafter referred to as LSEDIT, is an interactive text editor designed specifically for software development. It assists the user in quick and accurate development of syntactically correct “source” code.

In this case, L^AT_EX or S^LT_EX are the “languages” being used.

2 LSEDIT Definitions

Placeholders are optional or required constructs which are usually inserted into the file you are editing as a part of a template. Placeholders represent places in the source code where the user must provide additional information. In some cases, a placeholder may be expanded to provide a template for this additional text.

Tokens are keywords that may be typed anywhere into the file you are editing and expanded to provide a template for a corresponding construct.

3 LSEDIT Commands

The following key commands are the most commonly used. They include:

Command	Meaning
<code>CTRL/E</code>	Expand the current placeholder or token
<code>GOLD</code> + <code>CTRL/E</code>	Un-expand the current placeholder or token
<code>CTRL/K</code>	Delete the current placeholder
<code>GOLD</code> + <code>CTRL/K</code>	Un-delete the current placeholder
<code>CTRL/N</code>	Go to the next placeholder
<code>CTRL/P</code>	Go to the previous placeholder
<code>GOLD</code> + <code>HELP</code>	Get help for the current placeholder or token

4 Conventions Within The L^AT_EX Environment

There are two types of placeholders: 1) *required*, and 2) *optional*. They are denoted within the L^AT_EX environment as follows:

Symbol(s)	Meaning
<code>< ></code>	Single required placeholder
<code>< > . . .</code>	List of required placeholders
<code><< >></code>	Single optional placeholder
<code><< >> . . .</code>	List of optional placeholders

5 Invoking LSEDIT

5.1 Accessing the L^AT_EX Environment

You can invoke LSEDIT and access the L^AT_EX environment in one of two ways:

1. You can give the /ENVIRONMENT qualifier on the command line. For example,

```
$ LSEDIT/ENVIRONMENT=LSE_ENVIRONMENTS:LATEX file.TEX
```

where *file* is the name of your input file. The file type **must** be .TEX or .STY.

2. You can define a logical symbol LSE\$ENVIRONMENT:

```
$ DEFINE LSE$ENVIRONMENT LSE_ENVIRONMENTS:LATEX
```

Then you may omit the /ENVIRONMENT qualifier from the command line when you invoke LSEDIT. This logical may be a search list.

Either method will accomplish the same result, although the second method will reduce the number of keystrokes and will assure that you don't forget the qualifier when you enter LSEDIT.

Note: The examples above assume there is either a system or process logical LSE_ENVIRONMENTS that points to where LSEDIT environments are kept.

6 Supported L^AT_EX Document Styles

When you invoke LSEDIT to create a new L^AT_EX file, the screen will have the initial placeholder <LaTeX> on the first line of the buffer. When you expand this placeholder by pressing CTRL/E, a menu will be displayed listing the different L^AT_EX styles that are supported by the environment. The document styles supported are:

- **article** – standard style
- **book** – standard style
- **letter** – standard style
- **report** – standard style
- **slides** – SL^IT_EX

The style is chosen by using the arrow keys to point to the desired style and pressing RETURN.

7 Running L^AT_EX From Within LSEDIT

Your L^AT_EX source file can be processed through L^AT_EX *without* leaving LSEDIT by using one of two methods.

7.1 Using Standard LSEDIT Commands

When you are finished editing, press the Do key or `[GOLD] + [KP 7]` and enter

```
LSE Command> COMPILE
```

LSEDIT will write out your buffer and spawn a subprocess to run L^AT_EX. You will receive a completion message when the subprocess finishes.

You can then read in the .LIS file to check for errors, fix your source and start again.

If no errors occur, the subprocess will report that the “compilation” is complete. At this point, you can exit LSEDIT and run ATEX¹, or you can run ATEX from within LSEDIT by pressing `[GOLD] + [KP 7]` and entering

```
LSE Command> SPAWN ATEX FOOBAR
```

7.2 Using the L^AT_EX Review System

When you have finished editing, press `[GOLD] + C`. This will spawn a subprocess to “compile” your file. When the job is finished, it will automatically open a review window with the .LIS file.

While inside a L^AT_EX review, the following key definitions are active:

- `[CTRL/F]` : Next error
Positions to and highlights the next error message (following the current position in the review buffer, not necessarily the last error message you viewed). If there are no more error messages in the buffer, the editor will display the message “No more errors”. This function mimics the `NEXT ERROR` command in an ordinary LSE review session.
- `[CTRL/B]` : Previous error
Moves backwards through the review buffer. If there are no more error messages between the current position in that buffer and the start of the buffer, the message “No more errors” is displayed. This function mimics the `NEXT ERROR` command in an ordinary LSE review session.
- `[CTRL/G]` : Goto source
Attempts to position the source buffer to the line corresponding to the currently highlighted error message. The strategy is to search the source

¹ATEX is SI's command procedure which invokes the DVI-to-device-specific-output translator and initiates printing; the appropriate equivalent for your site should be substituted here.

buffer for a line which matches the last source line in the review buffer preceding the current error message. This function mimics the `GOTO SOURCE` command in an ordinary LSE review session.

- `CTRL/E` : End \LaTeX review

You can also review a \LaTeX file that has already been “compiled” by entering the `LTREV` (\LaTeX review) command as follows:

```
GOLD + KP 7 or Do
LSE Command> LTREV
```

This command will do the following:

1. Split the screen (if not already done).
2. Read into the `$REVIEW` buffer the associated `.LIS` file for the source.
3. Position to and highlight the first error message in the `.LIS` file.
4. Redefine the standard `LSEDIT` keys for review navigation as shown above.

The cursor will remain positioned at the start of the line containing the first error message. Note that the `LSEDIT` commands `NEXT ERROR`, `PREVIOUS ERROR`, and `GOTO SOURCE` are *not* redefined! You must use the keys listed above.

To finish the review (and to restore the previous definitions of your keys), enter the `LTEND` command:

```
GOLD + KP 7 or Do
LSE Command> LTEND
```

This removes the review buffer from the screen, restores the key definitions, and returns the screen to one-window mode. (The `GOLD + E` key during a review session does the same thing as the command `LTEND`.)

8 Accessing Help for Placeholders and Tokens

On-line help is available for most of the placeholders and tokens defined within the \LaTeX environment.

For placeholders, position the cursor at the placeholder by pressing `CTRL/N`. Then press `GOLD + HELP`. If there is no help for this placeholder, a message will be given stating that fact.

For tokens, type the token and press `GOLD + HELP`.

If the placeholder or token is a menu, you can access help for items in the menu by pointing at the menu item with the arrow keys and then pressing `GOLD + HELP`.

For a complete list of all defined placeholders or tokens, press the `Do` key or `GOLD + KP 7` and enter

LSE Command> SHOW PLACEHOLDER *

or

LSE Command> SHOW TOKEN *

A Example Page Layout

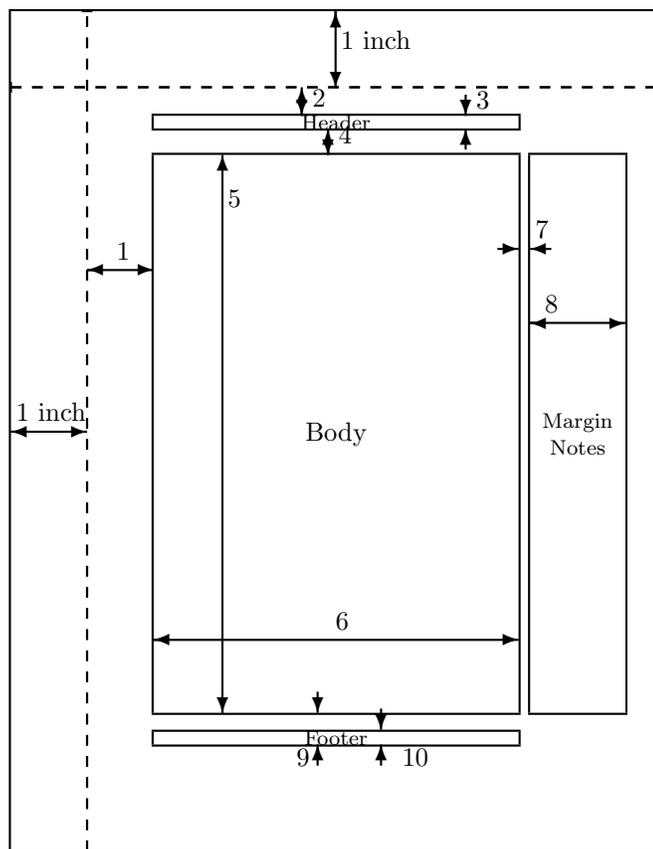
L^AT_EX refers to internal parameters to decide where headers, footnotes, margin notes, and the main text will be placed on the page. Figure 1 shows how L^AT_EX lays out a page for the one-column, single-sided `article` style using 10-point type.

Each of the parameters shown in the Figure 1 can be changed using the length commands such as `\setlength`, `\addtolength`, and `\settowidth`.

Note: You *must* change these parameters *before* the `\begin{document}` command.

The sizes for the other document styles and point sizes may be different. These style files are in the `TEX_INPUTS` directory. The file type is `.STY`. You can list or print any of these files to examine the page layout parameters.

The reference point for all L^AT_EX parameters is one inch down and one inch to the right of the upper left hand corner of the page.



1 \oddsidemargin = 63pt	6 \textwidth = 345pt
2 \topmargin = 27pt	7 \marginparsep = 11pt
3 \headheight = 12pt	8 \marginparwidth = 90pt
4 \headsep = 25pt	9 \footskip = 30pt
5 \textheight = 528pt	10 \footheight = 12pt

Figure 1: Example L^AT_EX Page Layout

B Example Type Styles

The following items illustrate the different type styles available and their declarations.

- `\rm`: Roman (default)
- `\em`: *Emphasis, toggle between roman and italics*
- `\bf`: **Bold**
- `\it`: *Italics*
- `\sl`: *Slanted*
- `\sf`: Sans serif
- `\sc`: SMALL CAPS
- `\tt`: Typewriter

C Example Type Sizes

The following items illustrate the different type sizes available and their declarations.

- `\tiny`: This is tiny text.
- `\scriptsize`: This is scriptsize text.
- `\footnotesize`: This is footnotesize text.
- `\small`: This is small text.
- `\normalsize`: This is normalsize text (default).
- `\large`: This is large text.
- `\Large`: This is Large text.
- `\LARGE`: This is LARGE text.
- `\huge`: This is huge text.
- `\Huge`: This is Huge text.