

A L^AT_EX Package for Typesetting EDSAC Programs

Lee Wittenberg
leew@alumni.stanford.edu

January 18, 2022

This document describes and demonstrates the use of `edsac.sty`, a (currently experimental) L^AT_EX package for typesetting EDSAC programs in a manner similar to that in *The Preparation of Programs for an Electronic Digital Computer** by Wilkes, Wheeler and Gill (1951). The package is loaded with the usual

```
\usepackage{edsac}
```

and it provides the `edsac` environment and a number of supporting commands. Package options are described below (Section 4).

1 The edsac Environment

EDSAC programs are typically typeset with a single order (or control combination) per line in 3 columns, separated by vertical rules: the location of the order, the order itself, and an optional comment. For example, a trivial program loaded at location 56 that does absolutely nothing:

		T	56 K		load from location 56
56		X	F		do nothing
57		Z	F		stop

would be specified in L^AT_EX as:

```
\begin{edsac}{56}
\nonum | \T 56 \K | load from location 56  \\
        | \X    \F | do nothing             \\
        | \Z    \F | stop
\end{edsac}
```

which resembles a typical `tabular` environment, except that a vertical bar (‘|’) is used instead of an ampersand (‘&’) to separate columns. In fact, a `tabular` environment is used to implement the `edsac` environment.

As you can see, each code letter is specified by the corresponding single-character macro (non-alphabetic characters are specified by macros based on the substitute characters used in Martin Campbell-Kelly’s simulator: ‘\@’ for ‘θ’,

*Hereinafter referred to as WWG.

‘\!’ for ‘ ϕ ’, ‘\&’ for ‘ Δ ’, and ‘\#’ for ‘ π ’). The perforator characters for “erase” and “blank tape” are specified by the macros ‘*’ and ‘\.’, respectively. The `\nonum` command is used for control combinations (which are not placed in the store), and the environment’s single argument specifies the location of the starting order.

2 A Second Example

Campbell-Kelly’s “Hello, World” program provides a slightly more complex example:

	T	64 K	Load from location 64
	G	K	Set θ parameter
Start \rightarrow 0	Z	F	Stop
1	O	5 θ	Letter shift
2	O	6 θ	Print ‘H’
3	O	7 θ	Print ‘I’
4	Z	F	Stop
5	*	F	Letters
6	H	F	‘H’
7	I	F	‘I’
	E	Z	} Enter at location 0 θ
	P	F	

and introduces the `\from` command to indicate the target of a transfer order and the ‘\|’ order column prefix to indicate constant values.

```

\begin{edsac}{0}
\nonum      |   \T 64 \K | Load from location 64   \\\
\nonum      |   \G   \K | Set $\theta$ parameter  \\\
\from{Start}|   \Z   \F | Stop                               \\\
            |   \O 5 \@ | Letter shift                \\\
            |   \O 6 \@ | Print ‘H’                     \\\
            |   \O 7 \@ | Print ‘I’                     \\\
            |   \Z   \F | Stop                               \\\
            ||| \*   \F | Letters                        \\\
            ||| \H   \F | ‘H’                             \\\
            ||| \I   \F | ‘I’                             \\\
\nonum      |   \E   \Z | \longnote{2}{10em}
                                     {Enter at location $0\theta$} \\\
\nonum      |   \P   \F |
\end{edsac}

```

The environment also provides the `\multifrom` command for when an order is targeted by multiple E/G orders, as in subroutine C7 (WWG, pp. 118–20). The order numbers in the `\multifrom` argument are separated by ‘\’s. For example, line 34 of C7 would be typeset with “`\multifrom{30\31}`.”

This example also introduces the `\longnote` command (which can be used outside of an `edsac` environment). It takes 3 arguments: (1) the number of rows the note extends over, (2) the width to allow for the note, and

(3) the note itself. For a note that must extend onto a following page, an optional argument, `[tn]` or `[bn]`, may specify the number ($n < 1^{\text{st}}$ arg) of commented lines on the top or bottom, respectively, of the current page.

3 More Features

The `edsac` environment also provides the order column prefixes ‘`_`’ and ‘`\(`’. The former underlines the order it introduces, and is used to indicate a transfer of control. The latter parenthesizes the order, and is used to indicate that an order will be changed during execution. Examples of both are found in the code for Section 5, as are examples of the badly-named (and not-particularly-well-implemented) `\eindent` construct, used to keep the columns aligned in consecutive `edsac` environments.

The ‘`\#`’ construct is typeset appropriately whether used as a pseudo-order or to modify a code letter. For example, the code

```
| \#      \F | figure shift  \\  
| \A 26\#\@ | add long word at 26
```

would be typeset as

π	F	figure shift
A 26 $\pi\theta$		add long word at 26

Sometimes, as in the previous example and the WWG explanation (p. 22) of the “Wheeler jump,” the vertical rules are not wanted, and the starting address is a generic m or n . The rigid length `\erulewidth` controls the width of the vertical rules, and can even be set to `0pt`, if desired. If the `\elineprefix` command is defined at entry, the `edsac` environment generates line numbers with the specified prefix added to all non-zero line numbers. These commands are automatically undefined at the end of the environment so that any program listings that follow will be typeset in the default format.

Number of storage location		Order	
m	A	m	F
$m + 1$	G	n	F

The orders in the subroutine are as follows:

	G	K	
n	A	3	F
$n + 1$	T	$p + 2$	θ
$n + 2$			
\dots			
$n + p + 1$			
$n + p + 2$	(Z		F)

} subroutine operational orders
(p in number)

In the corresponding L^AT_EX code, which follows, note the use of the standard tabular `\multicolumn` command, which works as usual, to interject non-program material into the listing. Note also the use of `\eresetline` to adjust line numbers and the plain T_EX command `\global\def` to redefine `\startloc` within the environment (because L^AT_EX's `\renewcommand` only works locally, within the current column).

```

\setlength{\erulewidth}{0pt}
\newcommand{\startloc}{m}
\newcommand{\elineprefix}{\ensuremath{\startloc}}
\begin{edsac}{0}
\multicolumn{1}{c}{Number of}          \\\
\multicolumn{1}{c}{storage}            \\\
\multicolumn{1}{c}{\uline{location}}    \\\
| \multicolumn{1}{c}{\uline{Order}}    \\\
| \A $m\enspace$ \F                    \\\
| \G $n\enspace$ \F                    \\\nonum\\
\multicolumn{3}{l}{The orders in the subroutine
are as follows:}
\\\nonum\global\def\startloc{n}\eresetline\\
\nonum | \G \K \\\
| \A 3 \F \\\
| \T $p+2$ \@ \\\
| | \longnote{3}{.4\textwidth}
| | {\hang subroutine operational
orders\\($p$ in number)}\\
\multicolumn{1}{c}{\ensuremath{\ldots}}
\global\def\startloc{n+p}\eresetline[1]\\
| \\\
| \C \Z \F
\end{edsac}

```

Finally, similar to `\elineprefix`, if `\elinesuffix` is defined prior to an `edsac` environment it adds the specified code at the end of each line number. This is rarely needed, but is useful in situations, such as in library subroutine A3, where integers are placed directly in the code:

Part I:	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> R9 modified </div>	order 3 in R2 is altered to S 40 D. Thus the integers are placed in the store neg- atively and become $-10^{-n}, n = 0 \dots 7$
	T $60\pi\theta$ K	
$60\pi\theta$	17 179 869 184 F	
$62\pi\theta$	1 717 986 918 F	
$64\pi\theta$	171 798 692 F	
$66\pi\theta$	17 179 869 F	
$68\pi\theta$	1 717 987 F	
$70\pi\theta$	171 799 F	
$72\pi\theta$	17 180 F	
$74\pi\theta$	1 718 π	

In addition to the use of `\elineprefix`, the \LaTeX code has a number of interesting features. It makes great use of the `\multicolumn` command (the `\mc` definition is necessary because a ‘|’ in a preamble option causes problems when used within an `edsac` environment). The `\skipword` command (in `\num`) increments the line number counter for each long word value.

```

\newcommand{\elinesuffix}
  {\makebox[0pt][l]{\ensuremath{\pi\theta}}}
% avoid conflict with | in edsac environment:
\newcommand{\mc}{\multicolumn{1}[l]{Part I:}}
\newcommand{\num}[1]
  {\multicolumn{2}{p{7.5em}}{\hfill#1}\skipword}
\begin{edsac}{60}
\mc
  | \multicolumn{1}{c}
    {\fbox{\parbox[t]{4em}{\centering R9\modified}}}
  | \parbox[t]{.5\textwidth}
    {\small\hang order 3 in R2 is altered to
      S~40~D. Thus the integers are
      placed in the store negatively
      and become  $-10^{-n}$ ,  $n=0\ldots 7$ } \\
\nonum | \multicolumn{2}[l]{T  $60\pi\theta$  K} \\
  | \num{17 179 869 184 F} \\
  | \num{ 1 717 986 918 F} \\
  | \num{   171 798 692 F} \\
  | \num{    17 179 869 F} \\
  | \num{     1 717 987 F} \\
  | \num{      171 799 F} \\
  | \num{       17 180 F} \\
  | \num{        1 718  $\pi$ }
\end{edsac}

```

4 Package Options

The package provides two options: `wwg` and `rulecolor`. The former uses square brackets (à la WWG) instead of curly braces for long comments, the latter specifies a color for rendering the rules in code listings. Options may be specified when the package is loaded (e.g., `\usepackage[wwg]{edsac}`) or using the `\edsacoptions` command. The following section is rendered using the options `wwg` and `rulecolor=red`.

5 A Fuller Demonstration

For a fuller demonstration, Section 7-3 from WWG (pp. 51–53) does the job nicely, and uses most of the techniques described above (see the \LaTeX source for details):

7-3 Alternative method for Example 2.

The example given in Section 7-2 will now be repeated in a revised form making use of assembly subroutine M1. The components of the program are

H sequence,
Master routine,
Auxiliary subroutine,
Library subroutines R9, Q2, P1, and D6.

R9, however, is not dealt with by M1 but is automatically placed in its usual position (56–70). The H sequence consists of a number of pseudo-orders which, in Section 7-2 were included at the end of the master routine.

Storage space is allocated as follows:

56–70	R9		
71–75	unused		
76	reference order for master routine		
77	do.	auxiliary] See Section 4-62
78	do.	Q2	
79	do.	P1	
80	do.	D6	
82–97	M1		
98–	H sequence		

7-31 Make up of tape.

R9	R9 begins with P K T 56 K, so that it is placed in locations 56–70.
----	---

space

P K T 82 K	First order of M1 goes into 82.
------------	---------------------------------

M1	
P 76 F	Reference order of master routine goes into 76.
T 98 K	First order of H sequence goes into 98.

] parameters
used by
M1

space

P Z G K	
E 82 K T F	Calls in M1, which places P 98 F in 45.

H sequence	Placed with first order in 98.
------------	--------------------------------

space

P Z G K

T 92 K T ϕ

E 82 K I F

Sets M1 ready to deal with master routine.
Calls in M1, which places reference order
in 76.

Master routine

space

P Z G K

E 82 K P F

Calls in M1, which places reference order
in 77.

Auxiliary

space

P Z G K

E 82 K P F

Calls in M1, which places reference order
in 78.

T 45 F

P 72 D

P 1 ϕ

Plants parameters required by
Q2.

Q2

space

P Z G K

E 82 K P F

Calls in M1, which places reference order
in 79.

P1

P Z G K

E 82 K P F

Calls in M1, which places reference order
in 80.

D6

E 25 K

E ϕ P F

Sends control to the first
order of the master routine.

7-32 H sequence.

H	0	Δ	F	line feed
	1	θ	F	carriage return
	2	K 2048	F	$15 \cdot 2^{-4}$
	3	R	F	$4 \cdot 2^{-4}$
	4	E	F	$3 \cdot 2^{-4}$
	5	I	F	$8 \cdot 2^{-4}$
	6	π	F	figure shift

7-33 Master routine.

		G	K	
Start \rightarrow	0	O	6 H	figures
	1	O	1 H	carriage return
	2	O	H	line feed
	3	T	72 D] Sets limits of integration: 0 to 72D, 1/2 to 74 D.
	4	A	5 H	
	5	T	74 D	
	6	A	6 θ	
	7	G	2 ϕ] calls in Q2
Q2 \rightarrow	8	A	8 θ] calls in P1
	9	G	3 ϕ	
	10	\parallel P	10 F	parameter for P1
P1 \rightarrow	11	O	6 H	extra output order to print last figure
	12	Z	F	

7-34 Auxiliary subroutine

		G	K	
	0	A	3 F	
	1	T	15 θ	plant link
	2	H	D	
	3	V	D	x^2 to accumulator
	4	A	3 H	$4 \cdot 2^{-4} + x^2$
	5	Y	F	
	6	T	D	$0.25 + x^2$ to 0D
	7	H	2 H	$15 \cdot 2^{-4}$
	8	V	D	
	9	Y	F	
	10	T	4 D	$15/16 \cdot (0.25 + x^2)$ to 4D
	11	A	4 H	
	12	T	D	3/16 to 0D
	13	A	13 θ] Calls in D6, which places the integrand in 0D.
	14	G	4 ϕ	
D6 \rightarrow	15	(Z	F)	link

Known Issues

- The `\eindent` command is very badly named (and its implementation is somewhat problematic). This is also true for a number of the other commands. Suggestions for improvement (both in naming and implementation) would be greatly appreciated.
- It is technically possible to have the characters ‘P’, ‘Q’, etc. declared “active” and use them as-is (i.e., without a backslash) in the order part of a line. However, my `TeX`pertise doesn’t extent that far (and I’m not completely sure it’s a Good Idea). Any help in this area would be welcomed.
- The `edsac` environment argument really should be optional, with zero as the default. However, `LATEX` won’t recognize ‘|’ as a column separator when it is looking for the ‘|’ that would introduce the optional argument value. I couldn’t figure out a clean way to deal with this. A non-optional argument seemed the simplest solution.
- WWG listings “stretch” the arrow that would be used in our `\from` command. Can anyone suggest how to do this in `LATEX`?
- The `edsac` environment is based on `tabular`, and therefore, cannot extend past the end of the page. It would be nice if `supertabular` or `longtable` could be used instead. However, I wasn’t able to accomplish this.
- All dimensions used in the package are based on the characteristics of a document’s primary font. Therefore the defined commands *should* work correctly regardless of typeface and type size. However, it has only been minimally tested—only with Computer Modern and Times Roman at 10, 11 and 12-point sizes.

Any bug reports (and fixes) or suggestions for improvement should be addressed to the author at `leew@alumni.stanford.edu`. Even if you find no bugs, please let me know if you find this package useful (so I can let you know about any updates and modifications).