NAME

s2latex − convert Scribe input files to TeX/Latex input files

SYNOPSIS

s2latex [**−k**] [**−q**] [**−s** symfile] file ...

DESCRIPTION

S2latex reads the *files* and produces a *Latex* input file on the standard output. If no files are given, s2latex reads standard input.

-k outputs every unrecognized Scribe keyword to stderr. Default translation rules are applied to unrecognized keywords (i.e., *@keyword(...) → \keyword{...}* and *@begin(env) → \begin{env}).* This option can be used to call your attention to keywords that should use other translation rules.

-q changes the processing of double quotes. Normally, double quotes are converted to directed single quotes (i.e., "..." → ‘...’). This flag leaves the double quotes (they are printed from the "tt" font).

-s gives the name of a file containing additional or replacement translation rules for Scribe keywords or environments. Each line of this file should have the format:

word rule replacement

word is the keyword or environment name (a keyword name is distinguished by a leading "@"). This name should be given in **lower case .** *Rule* is the name of the translation rule (see below). It can be in either case and only the first character is significant. *Replacement* is replacement text for the keyword. It must be given even if not required by the rule.

S2latex has 9 translation rules (read "→" as "is converted to"):

replace:
@kw(...) → \rep{...}
@begin(env)...@end(env) → \begin{rep}...\end{rep}
I.e., the keyword or environment name is simply replaced by a new name.

delete: *@kw(...) → ...*
@begin(env)...@end(env) → ...
I.e., the keyword or environment is deleted but the enclosed text is left.

font: *@kw(...) → {\rep ...}*
@begin(env)...@end(env) → {\rep ...}
This is the appropriate translation for a TeX font; i.e., @i(xyz) to {\it xyz}.

environment:
@kw(...) → \begin{rep}...\end{rep}
@begin(env)...@end(env) → \rep{...}
I.e., a keyword is converted to an environment or an environment to a keyword.

itemize:
@kw(...) → \begin{rep}...\end{rep}
@begin(env)...@end(env) → \begin{rep}...\end{rep}
In addition, the start of each new paragraph in the enclosed text will be preceded by a "\item ".

tag: (aka, "description"). Like *itemize* but the start of each new paragraph is preceded by "\item[" and the first tab in the paragraph is replaced by "] ". I.e., this is the Latex "description" environment.

comment:
The keyword or environment and all of the text enclosed are output preceded by "%" (the TeX comment character). I.e., this "comments out" a keyword or environment.

alignment:
@kw(...) → \begin{rep}...\end{rep} @begin(env)...@end(env) → \begin{rep}...\end{rep}
Each Scribe tab in the enclosed text is replaced with " & " (the TeX alignment tab) and each newline is preceded by "\\".

nop: *@kw(...) → \rep{...}*
Like *replace* but has the side effect of saying this keyword generates no text. (This is used when s2latex is trying to identify the "tag" portion of a paragraph in a @description environment).

S2latex starts with the following translation rules. The user can augment or redefine any of these rules via the **’-s’** command line option.

@unnumbered	REPLACE	chapter*
@bigsection	REPLACE	chapter
@blankspace	REPLACE	vspace
@ux	REPLACE	underline
@example	ENV	example
@center	ENV	center
@i	FONT	it
@u	FONT	it
@un	FONT	it
@b	FONT	bf
@p	FONT	bf
@t	FONT	tt
@c	FONT	sc
@r	FONT	rm
@w	FONT	obeyspaces
@itemize	ITEMIZE	itemize
itemize	ITEMIZE	itemize
@enumerate	ITEMIZE	enumerate
enumerate	ITEMIZE	enumerate
@description	TAG	description
description	TAG	description
@multiple	DELETE	-
multiple	DELETE	-
@format	ALIGN	tabular
format	ALIGN	tabular
@comment	COMMENT	comment
comment	COMMENT	comment
@index	NOP	index
@tabs	NOP	tabs

SEE ALSO

tex(1), latex(1)

AUTHOR

Van Jacobson, Lawrence Berkeley Laboratory (van@lbl-csam.arpa) with thanks to Richard Furuta for much help decoding Scribe.

BUGS

The program isn’t intended to do a 100% conversion. It usually gets about 90% right on the first pass. With the addition of a few translation rules & some judicious editting of the Scribe input file, it can get you 99% of the way to a document that Latex will accept. This may still be a long way from a document that you’ll accept but such are the joys of typesetting.

The "parse error at line xxx" message isn’t all that helpful. Start by looking in the input file around the line number given. Errors are usually caused by:

- a Scribe macro or environment definition. Delete it from the input file & try again.
- some interesting new way to do "descriptions" or "formats". If it’s a description, either fix things so s2latex can find the tag or put in a dummy tag so that the rest of the description environment doesn’t get screwed up. If it’s a format, either the fixup will be obvious or you should change to format keyword to something unrecognized & fix things later in the output file.

The following two character control sequences probably should be converted to something but aren’t: @!, @$, @:, @/, @|, @&, @), @# and @].

In a format environment, all Scribe tab commands (@\, @=, @>, @^) get converted to the TeX alignment tab (&).

A lot better job could be done on formats and macro definitions.