NAME

espp − A preprocessor for the Eiffel source pretty printer

SYNOPSIS

espp [**−lnsux**] [**−t | −2 | −3 | −4**] [**−ascii | −ansi | −html | −latex | −mif | −mime | −rtf**] **<files...>**

DESCRIPTION

espp is a *perl* script that works as a preprocessor for *espretty* the (Eiffel source pretty printer). It can be used to pr
'pretty-printed' versions of Eiffel class text files and thus make these easier to read.

The *espp* preprocessor shell script calls the *espretty* program, which understands a substantial amount about the syntax o
fel, but which doesn't make any attempts to cope with incomplete and misformed syntax. However, whenever possible it t
format the class text according to the layout standards in ETL Appendix A.

espp parses the given options and class file arguments and passes these to relevant *espretty* calls. The main purpose of thi
preprocessor version for the *SmallEiffel* compiler is, to pass multiple file arguments to *espretty* calls and to *create* alway
files instead of using STDOUT.

espp gives you the ability to handle multiple input files at once, which the *espretty* program by itself doesn't support.

espp has the same options as *espretty* but allows the handling of multiple input files on the command line.

However, the preprocessor's execution can be interrupted at any time by pressing the *BREAK key* (Ctrl-C) under most Uni
tems.

OPTIONS

−t | −2 | −3 | −4

Indent levels by *NUM* blank characters, the *default* is *3.* This conforms to the code examples shown in ETL
NUM argument should be one of those listed above. Use *−t* (8 blanks) to emulate a tabstop. However, *espp* d
check for other values, so it is the responsibility of the user to provide one of those *NUM* arguments shown above

−ascii | −ansi | −html | −latex | −mif | −mime | −rtf

Use one of these formatters for output, the *default* is *ASCII.* Available formatters are:

ASCII

Text is printed using plain ASCII with no embellishments. The new created (formated) files will have the exte
**.e* and the backups (processed original files) the extension *.bak.

ANSI

As ASCII, but with keywords emboldened with the relevant ANSI escape sequences. For use if you have a ter
that supports ANSI bold codes under Unix, or under DOS if have included the ANSI.SYS driver in your DOS
FIG.SYS file. This format is supported by Unix pagers like *more , less or most.* The new created files will ha
suffix of the formater appended, in this case **.ans* and the original files are kept with the "*.e" extension.

HTML

Outputs the text including a minimal subset of HTML sequences. This format is recognized by WWW-browse
Netscape etc. and a lot of HTML-editors. With this formatter you should be able to present typeset Eiffel class
WWW. The new created files will have the suffix of the formater appended, in this case **.htm* and the original fi
kept with the "*.e" extension.

LaTeX

Outputs the text including a minimal subset of LaTeX sequences. This format is recognized by the LaTeX Doc Preparation System. With this formatter you should be able to present typeset Eiffel classes in LaTeX. The ne ated files will have the suffix of the formater appended, in this case **.tex* and the original files are kept with the extension.

MIF

Outputs the text including a minimal subset of FrameMaker Maker-Interchange-Format sequences. This for recognised by the FrameMaker publishing software. If you load the output you should be able to print typeset classes. The new created files will have the suffix of the formater appended, in this case **.mif* and the original fi kept with the "*.e" extension.

NOTICE: espretty's implementation of this formatter has some *restrictions* and *limitations in* the proper *Eiffe ment handling* , for more information about this see also the *CAVEATS section.*

MIME

Outputs the text including RFC 1341 Rich-Text sequences. If you have a MIME richtext reader or you wish t typeset Eiffel Classes to people, this formatter is for you. Files created with this formater will have the suffix formater appended, in this case **.mim* , the original files are kept with the "*.e" extension.

RTF

Outputs the text including a minimal subset of Microsoft's Rich-Text-Format sequences. This format is recogni a lot of WordProcessors. If you load the output you should be able to print typeset Eiffel classes. Files created wi formater will have the suffix of the formater appended, in this case **.rtf* , the original files are kept with the extension.

NOTICE: espretty's implementation of this formatter has some *restrictions* and *limitations in* the proper *Eiffe ment handling* , for more information about this see also the *CAVEATS section.*

−l When this option is set, Eiffel identifiers (not types) are transformed into lower chars, this conforms to the Appendix A layout guidelines.

−n Print output with line numbers and statement level, empty lines are ignored. This is not a bug, it's a feature. If bined together with the -x option, the output of empty lines without line numbers can be reduced.

−s Print output in a sort of short format, removes feature bodies.

−u When this option is set, (most) Eiffel types are transformed into upper chars, this might reduce the amount of s caps-lock presses under your keyboard.

−x Start new line after 'redefine, until ...', removes empty lines in routine bodies.

EXAMPLE USAGE

espp can process multiple class files as arguments (thanks to the Unix shell expansion mechanism). For example:

```
espp class1.e class2.e class3.e...
```

or for lazy typists:

```
espp *.e
```

For documentation purposes to get a short form in RTF format:

```
espp -s -rtf  class1.e class2.e class3.e...
espp -s -rtf  *.e
```

Or:

```
espp -t -u -n -mif class1.e class2.e class3.e...                    espp -t -u -n -mif *.e
```

Of course there are much more possibilities, try them out.

SEE ALSO

espretty (1), *unformat (1)*

[ETL92] **Bertrand Meyer, Eiffel: The language, Prentice Hall**

CAVEATS

The *RTF* , *HTML* , *LaTeX* , and *MIF* output formatters have some *limitations* in the *Eiffel comment* handling. So the u
forced to take care about the following *restrictions* when these formatters are used:

Eiffel names of ‘features’ or other ‘entity’ appearing in a comment *must* be enclosed in *exactly* these single quotes (one *op*
quote -> ‘ <- and one *closing quote* -> ’ <-), as shown here with *‘feature’* and *‘entity’* , to be printed correctly in *italics.*

Additional you have to *avoid generally* using this sort of single quotes for other purposes in Eiffel comments. This is *imp*
for the *RTF* , *HTML* , *LaTeX* , and *MIF* formatters, to produce a correct output format, otherwise the output of their f
sequences might not be correct.

Ending comments of *routines* and *classes* can not be printed in *italics* until now with this version for *SmallEiffel.* S
doesn’t conforms to the ETL Appendix A layout guidelines. This might be corrected in a future release.

A correct example:

```
feature

  is_break (ch: CHARACTER): BOOLEAN is
                    -- Check if break-character is ‘ch’ <--- (correct
    do                                    quotes)
      Result := true
    end -- is_break
```

A wrong example:

```
feature
                                   (wrong quotes)
  is_break (ch: CHARACTER): BOOLEAN is          |
      -- Check if break-character is ’ch’ <----
      -- Control characters aren’t allowed for ‘ch‘
    do                   ˆ           ˆ
      Result := false    |           |
    end -- is_break      |______ (wrong quotes)
```

BUGS

espretty has sure some bugs, the *espp* perl script is (as far as I know) bugfree.

AUTHOR

Valentino Kyriakides