READ_ME for CLIP/UNIX

1 Directory CLIP/UNIX

The directory contains the files

a. read_me.ps, read_me.txt (this file).
b. clip.pas, clip, CLIP_MSG (CLiP system).
c. cl.csh, clp.csh, clp_env.csh, (user support).
d. ex01_a.ps, ex01_a.asc, ex01_a.ini, ex01_a.pas,
   ex01_b.ps, ex01_b.asc, ex01_b.ini, ex01_b.pas (example).
e. ada.ini, c.ini, fortran.ini, lisp.ini, modula.ini, pascal.ini (template INI-files for some languages).
f. manual.ps, manual.txt (manual).

About file extensions:

PS-files are postscript files and represent the type-set versions of the corresponding ASC or TXT-files.

ASC-files are derived by ASCII-export from the original word-processor files (in our case Lotus Manuscript).

TXT-files are ASCII versions of the corresponding PS-files. They are slightly modified to improve the lay-out. TXT-files are a service for people who have no easy access to a postscript printer.

2 Installing CLiP

a. Use FTP to copy the CLiP system to your local site. Note that clip is our own executable version of clip.pas, and it should be FTP-ed in binary mode.
b. Create a directory for the CLiP-system. This directory will be referred as the CLIP_DIR. Let us assume this directory is

   ~/clip

c. Copy clip.pas to your CLIP_DIR and compile. Since it is written in ISO-Pascal there should be no problem. We use the Sun Pascal 2.0 compiler under operating system SunOS 4.2 running on Sun4c (Sun1, Sun1+, Sun2) hardware. If you have problems you may want to try the binary version of clip.pas, clip, instead.
d. Copy the files CLIP_MSG, cl.csh, clp.csh and clp_env.csh to your CLIP_DIR. Make sure the file name CLIP_MSG is in uppercase.
e. Make the script files executable through the command

```
% chmod  +x  *.csh
```

3 A quick installation test

To test if your CLiP system has been properly installed, perform the following steps:

a. Choose a test directory. Let us assume ~/test.
b. Copy the files ex01_a.asc, ex01_b.asc, ex01_a.ini, ex01_b.ini, ex01_a.pas and

ex01_b.pas to your test directory.

c. Type the command

    % cl ex01_a.ini

  to produce the file palindrome.pas. Note that ex01_a.ini locates the source files as well as the target modules in the current directory.

d. Compare the file palindrome.pas with ex01_a.pas by means of the command

    % diff palindrome.pas ex01_a.pas

  There should be no differences.

e. Now type the command

    % cl ex01_b.ini

  to produce a new file palindrome.pas.

f. Compare the new palindrome.pas with ex01_b.pas by means of the command

    % diff palindrome.pas ex01_b.pas

  Again there should be no differences.

4 What you should know

clip performs the actual module extraction on the basis of a special file, CLIP.INI, which tells it literally everything it has to know in order to perform a run. Thus CLIP.INI specifies to clip:

- the files it has to read (i.e. the source files),
- the modules it should extract (i.e. target modules),
- the report file,
- the syntax of the CLiP-lines (i.e. the lines which trigger the module extraction process).

CLIP.INI is a straight forward ASCII-file. It has a simple structure which is explained in section 6 of the manual. In extreme situations you may want to edit a CLIP.INI manually but in general it will be more convenient to use the procedures cl.csh, clp.csh and clp_env.csh. These routines construct a CLIP.INI for you and run clip immediately afterwards. They are adequate most of the time.

Whenever you decide to create an INI-file to accommodate your special wishes, make sure you start from an INI-file you already know to be correct. Also if you want to change the syntax CLiP recognizes, please study section 2 of the manual.

A more extensive description of the CLiP system can be found in [1].

5 Using CLiP ...

The batch procedures below are designed to make your working directories independent of the CLIP_DIR. Remark that the original batch files assume that your CLIP_DIR is the directory user/clip.

5.1 ... via clp.csh and clp_env.csh

These procedures if one wants to extract all modules from up to 8 source files which are passed as parameters. First use clp_env.csh to define explicitly the source file directory and the target module directory, for example

    % clp_env.csh /user/test /user/test

defines the subdirectory test from the directory user as directory for the source files as well as for the target modules. With this setting the command

    % clp.csh file1.asc file2.asc file3.asc

will process the files /user/test/file1.asc etc. and extract all modules it encounters towards the directory /user/test.

All setting are assumed by the batch procedure. Originally it is tailored to Pascal programs of the sort displayed in the example ex01 (i.e. the files ex01_?.*) on the directory clip. To adjust clp.csh to another syntax one should be familiar with section 2 of the manual. Only modify the following lines clp.csh for this purpose:

line 15: Replace (* with the COMM-START of your choice.
line 16: Replace *) with the COMM-END of your choice.
line 17: Replace * with the CLIP-CHAR of your choice.
line 18: Replace ENDOF with the END-STRING of your choice.
line 19: Replace # with the OPTION-MARKER of your choice.
line 22: Replace clip.rpt with the report file name of your choice.

Note that the characters "a"-"z", "A"-"Z", "0"-"9" and "." do not qualify as characters for COMM-START, COMM-END, CLIP-CHAR or OPTION-MARKER while the END-STRING should be chosen from letters only.

5.2 ... via cl.csh

cl.csh is the fastest way to run clip and will probably be invoked most often. It can be used whenever an INI-file already is available. It copies the information of the specified INI-file to CLIP.INI and runs the clip program.

The INI-file to be used is passed as an optional parameter. If no INI-file is specified, cl.csh simply assumes CLIP.INI. For example

```
% cl ex01_a.ini
```

will copy the file ex01_a.ini to CLIP.INI and run clip. However the command

```
% cl
```

immediately runs clip with the assumption that a CLIP.INI already exists.

6 Specimen INI-files

The files ada.ini, c.ini, fortran.ini, lisp.ini, modula.ini and pascal.ini suggest a suitable CLiP syntax for a start. To customize clp.csh for such a syntax, refer to the previous section and to section 6 of the manual.

7 References

Ammers E.W. van, M.R. Kramer (1993), The CLiP Style of Literate Programming (submitted for publication). Anonymously ftp-able as CLIP_STYLE.PS, CLIP_STYLE_A.PS and CLIP_STYLE_B.PS from directory CLIP on sun01.info.wau.nl.

Table of Contents