READ_ME for CLIP/MS_DOS

1 Directory CLIP/MS_DOS

The directory contains the files

a. READ_ME.PS, READ_ME.TXT (this file).
b. CLIP_1.EXE, CLIP_2.EXE, CLIPHELP.HLP, CLIP_MSG.DAT (CLiP system).
c. CL.BAT, CLP.BAT, CLIP.BAT, CLP_ENV.BAT (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).

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_1.EXE, CLIP_2.EXE and CLIPHELP.HLP 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
   C:\CLIP
c. Copy the files
   CLIPHELP.HLP, CLIP_1.EXE, CLIP_2.EXE and CLIP_MSG.DAT
   to the CLIP_DIR.
d. Copy the files
   CL.BAT, CLP.BAT, CLIP.BAT, CLP_ENV.BAT
   to the directory that contains your batch files (this directory is assumed to be in the PATH definition of the AUTOEXEC.BAT).
e. If the name of your CLIP_DIR differs from the name "C:\CLIP" as suggested in point 2.a, the batch files
   CL.BAT, CLP.BAT and CLIP.BAT
   have to be adjusted. That is the string "C:\CLIP" has to be replaced by a string representing the correct name of your CLIP_DIR.

3 A quick installation test

To have a partial test (it does not check CLIP_1) to see if your CLiP system has been properly installed, perform the following steps:

a. Choose a test directory and make it current. Let us assume "C:\TEST".
b. Copy the files EX01_A.ASC, EX01_B.ASC, EX01_A.INI, EX02_B.INI, EX01_A.PAS and EX01_B.PAS to this test directory.
c. Type the command
   >CL EX01_A.INI
   to produce the file PALINDRO.PAS. Note that EX01_A.INI specifies the current directory for the source files as well as for the target modules.
d. Compare the file PALINDRO.PAS with EX01_A.PAS by means of the command
   >COMP PALINDRO.PAS EX01_A.PAS
   There should be no differences.
e. Now type the command
   >CL EX01_B.INI
   to produce a new file PALINDRO.PAS.
f. Compare the new PALINDRO.PAS with EX01_B.PAS
   >COMP PALINDRO.PAS EX01_B.PAS
   Again there should be no differences.

4 What you should know

The complete system executes in two passes.

The second pass, CLIP_2.EXE, performs the actual module extraction. It is guided by a file, CLIP.INI, which tells it literally everything it has to know in order to perform a run. Thus CLIP.INI specifies to CLIP_2.EXE:

- 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. Its fairly simple structure is explained in section 6 of the manual.

The responsibility of the first pass of the CLiP-system, CLIP_1.EXE, is purely to construct an appropriate INI-file for the task you want CLIP_2 to perform. More often than not the first pass can be short-cut and several batch files are supplied for this reason (see section 5).

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 "C:\CLIP".

5.1 ... via CLIP.BAT

CLIP.BAT performs the complete cycle to define and execute a run of CLiP. Thus both passes of CLiP are executed by this command. The procedure gives the user complete control over the extraction process and supplies context sensitive HELP when asked. The procedure works most comfortable when a mouse is available.

PROJECT menu allows your to LOAD an already existing INI-file or to SAVE the one you have defined so far. In addition you can start the extraction process from here.

SETTINGS/SOURCEFILES allows you to define the sources files that CLiP has to include in the extraction process. The source files may reside in different directories.

SETTINGS/TARGETMODULES lets you specify which modules you want to be extracted. The names of modules you put in the list are used either to identify the modules that have to be extracted (in which case you set the radiobutton EXTRACTED) or the modules that have to be omitted during the extraction process (in which case you set of course the radiobutton OMITTED). The default is OMITTED which of course implies that if no modules are listed, CLiP will extract every module it finds.

SETTINGS/PARAMETERS refers partly to non implemented features. Therefore don't touch the CLIP_MOD radiobuttons and forget about the INTERACTIVE FAULT CORRECTION. If you want error messages to be filed, you have to specify the report file in this menu.

SETTINGS/SYNTAX allows you to change the appearance of the lines that trigger the CLiPping process. By LOADing an appropriate INI-file (like PASCAL.INI, C.INI, MODULA.INI or FORTRAN.INI) you have an adequate suggestion for that particular language. The choice SHOW SYNTAX displays the way CLiP lines will look.

5.2 ... via CLP.BAT and CLP_ENV.BAT

This procedure can be used in a situation where on wants to extract all the modules from up to nine source files. The source files are passed as parameters. CLP.BAT creates a CLIP.INI file and starts the CLIP_2 program afterwards.

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.BAT one should sufficiently understand the syntax of CLiP [Ammers 1993]. Only modify the lines of CLP.BAT as follows

line 16: Replace "(*" with the COMM-START of your choice.
line 17: Replace "*)" with the COMM-END of your choice.
line 18: Replace "*" with the CLIP-CHAR of your choice.
line 19: Replace "ENDOF" with the END-STRING of your choice.
line 20: Replace "#" with the OPTION-MARKER of your choice.
line 23: 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.

If you are not sure of your modifications, you should compare the INI-file produced by CLP.BAT with the one that CLIP.BAT produces from the same set of preferences.

CLP.BAT must know the directory of the source files and the directory of the target modules. This information is taken from the environment variables CLIP_SRC and CLIP_TRG that can be set using CLP_ENV.BAT. Note that the directories must be specified excluding the trailing backslash. For example

```
>CLP_ENV  D:  C:\PROJECT\PASCAL
```

defines the home directory of drive D: as source file directory while the subdirectory PASCAL of directory PROJECT on drive C: will receive the extracted modules.

5.3 ... via CL.BAT

CL.BAT 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_2 program.

The INI-file to be used is passed as an optional parameter. If no INI-file is specified,

CL.BAT simply assumes CLIP.INI. For example

```
>CL EX01_A.INI
```

will copy the file EX01_A.INI to CLIP.INI and run CLIP_2. However the command

```
>CL
```

immediately runs CLIP_2 on the basis of an already existing CLIP.INI file.

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 such an INI-file run the CLiP system through CLIP.BAT. Then load the proper INI-file and modify the syntax by means of the SETTINGS/SYNTAX menu (see sect. 5.1).

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

1 Directory CLIP/MS_DOS 1

2 Installing CLiP 1

3 A quick installation test 2

4 What you should know 2

5 Using CLiP 2
5.1 ... via CLIP.BAT 2
5.2 ... via CLP.BAT and CLP_ENV.BAT 3
5.3 ... via CL.BAT 3

6 Specimen INI-files 4

7 References 4