READ_ME for CLIP/VMS

1 Directory CLIP/VMS

The directory contains the files

a. READ_ME.PS, READ_ME.TXT (this file).
b. CLIP_1.EXE, CLIP_2.EXE, CLIP_MSG.DAT (CLiP system).
c. USE_CLIP.COM, CL.COM, CLP.COM, CLP_ENV.COM, CLIP.COM, CLIP_1.COM (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 and CLIP_2.EXE 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 the name of this directory to be
   [USERNAME.CLIP]
c. Copy to your CLIP_DIR the files
   CLIP_1.EXE, CLIP_2.EXE, CLIP_MSG.DAT,
   USE_CLIP.COM, CL.COM, CLP.COM, CLP_ENV.COM and CLIP.COM
d. Adjust the procedure USE_CLIP.COM to reflect the actual CLIP_DIR you are using. This means you have to replace the string "[USERNAME.CLIP]" with a string that reflects you actual CLiP directory.
e. Make the commands CL, CLP, CLIP, CLP_ENV and CLIP_1 available by running the procedure USE_CLIP.COM. If you want to have CLiP available immediately after login, add the line
   @[USERNAME.CLIP]USE_CLIP
   to your LOGIN.COM.

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 this directory is [USERNAME.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 specifies the source files as well the target modules to be 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

   $DIFF PALINDROME.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 [USERNAME.CLIP].

5.1 ... via CLIP.COM

CLIP.COM performs the complete cycle to define and execute a run of CLiP. Thus both passes of CLiP are executed by this command. Probably you will not experience the first pass of CLiP as particularly user-friendly. But once you are familiar with its oddities it works fairly acceptable. Below you find some hints to make it work for you.

Main menu

The main menu presents 8 options. Option 1 and 8 need no further explanation. The remaining options are described below.

Option 2: Choose an initialization file

An INI-file is either build from scratch, or it is a modification of an already existing INI-file. Most likely you modify the INI-file of the last CLiP run you have made (4) or you start all over (2).

Option 3: Set run characteristics

This option refers mainly to future ideas to implement in CLiP and it doesn't work properly. You can safely skip this option in which case messages will be send to the display as well as to the file CLIP.RPT.

Option 4: Set list of source files

You can specify the source files CLiP has to scan by adding (1) file names to the list, deleting (2) file names from the list or changing (3) the order of the list. Don't forget to specify directories if the files are not in the current directory.

Option 5: Specify extraction of modules

By default the system assumes you want to extract all modules. If not you take menu item 1 to get into a submenu with three choices. Choice 3 lets you specify a list of modules which by choice 2 can be made to be either extracted or suppressed (omitted). Don't include directories with the specification of these modules.

Option 6: Save resulting initialization file

To save the run specification into a file, use option 6. It give you 2 menu items. The first specifies a name for the file (default CLIP.INI) and the second actually save the run specification into that file.

Option 7: Change syntax

This option allows you to adjust the syntax recognized by CLiP. Make sure you understand sufficiently about the operation of CLiP before fooling around here [Ammers 1993]. If not there is a fair change you end up with a syntax definition that is not consistent with your source files (there is danger though of destruction of files or the like).

Note that if you don't save explicitly through option 6, you simply quit the session and you loose the information you have specified.

5.2 ... via CLP.COM and CLP_ENV.COM

This procedure can be used in a situation where on wants to extract all the modules from up to eight source files. The source files are passed as parameters. CLP.COM 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.COM one should sufficiently understand the syntax of CLiP [Ammers 1993]. Only modify the lines of CLP.COM as follows

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.

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

CLP.COM allows you to identify different directories for the source files and the target modules. This information is taken from the logicals CLIP$SRC_DIR and CLIP$TRG_DIR which should be set using CLP_ENV.COM. For example

    $CLP_ENV [USERNAME.ASC] [USERNAME]

defines the subdirectory ASC of USERNAME as source file directory while the home directory of USERNAME will receive the extracted modules.

5.3 ... via CL.COM

CL.COM 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.COM 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.

5.4 ... via CLIP_1.COM

You will use this procedure if you want to create several INI-files at once. In this case you don't want to run CLiP after you have finished your session. The second pass of CLiP is not invoked.

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 option 7 (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