NAME

cpsdecode – Extract shell scripts and data files from Complete *PostScript* (CPS) file

SYNOPSIS

cpsdecode [*PostScript*-file] [**–n**] [**–v**]

DESCRIPTION

cpsencode [and **cpsdecode**] provide a convenient way of storing a UNIX script (or DOS batch file) and data files as part of the single *PostScript* plot the script creates. Thus, given the resulting Complete *PostScript* (CPS) file one can recreate the original script and data at a later time. **cpsdecode** reads a *PostScript* file (or stdin if none given) that previously has been augmented with output from **cpsencode**. It will then extract the embedded data files, undoing any uuencoding and compression that was performed by **cpsencode**.

–n Dryrun. No files are extracted but we report which files would be decoded|decompressed and saved to disk.

–v Verbose. Report the progress of decoding the script and any data files referred to in the script [Default is silent].

EXAMPLES

Let us say you have a cshell script called Figure_8.csh which creates the *PostScript* file Figure_8.ps. Figure_8.csh operates on several data files in order to make the plot. You turn this plot file into a Complete *PostScript* (CPS) file with the command

cpsencode –v Figure_8.csh >> Figure_8.ps

Alternatively, let that be the last command in the script so that it is automatically done by the script itself.
To unscramble the CPS file, simply say

cpsdecode –v Figure_8.ps

BUGS

cpsencode assumes good script etiquette so that any temporary files created by your script are removed before **cpsencode** is called. Otherwise they will take up unnecessary space in the Complete *PostScript* (CPS) file. **cpsencode** will have trouble if you script changes directories with the *cd* command since filenames become relative to another directory. Using absolute paths name for files (starting with /, ˜/. or ˜user/) is bad practice since other users may not be able to access those files.

AUTHOR

Paul Wessel, Geology & Geophysics, SOEST, University of Hawaii, 1680 East-West Road, Honolulu HI 96822, USA. www.soest.hawaii.edu/wessel.

NAME

cpsencode – Embed shell scripts and data files in Complete *PostScript* (CPS) file

SYNOPSIS

cpsencode shell-script [**–n**] [**–v**] [**–x**] >> *PostScript*-file

DESCRIPTION

cpsencode [and **cpsdecode**] provide a convenient way of storing a UNIX script (or DOS batch file) and data files as part of the single *PostScript* plot the script creates. Thus, given the resulting Complete *PostScript* (CPS) file one can recreate the original script and data at a later time. **cpsencode** takes a shell-script file as argument and writes the content of the script as well as the contents of any data files referred to in the script to standard output, which you can append to the *PostScript* file. This extra output is written after a logical EOF marker used by *PostScript* interpreters and begins with several comments flagged with the *PostScript* comment %%CPS. Script and data files are then written as *PostScript* comments. All data files are compressed with the external **bzip2** utility and converted to ASCII using a built-in uuencode algorithm.

–n Dryrun. No files are embedded but we report which files would be [compressed|encoded and] appended as *PostScript* comments.

–v Verbose. Report the progress of encoding the script and any data files referred to in the script [Default is silent].

–x Embed executable files. Default is to skip executable files as they are not expected to be portable. It is better to use the special mechanism (see below) to add in the source code of the program, as well as any makefiles or documentation needed.

SPECIAL

A special mechanism is available if you want to embed files that are not directly used by the script. These could be hidden files that some commands will read implicitly (say, a .cshrc file) or related documentation files that cannot be executed in the script. You can import such files by adding the script comment

in Unix shell scripts or

REM CPS: *filename*

in DOS .bat files.

You will need one such line for each filename you require (a single filename can contain wild cards, e.g., *.dat). Any file is only written once so repeated references to the same file by the script or this special mechanism will only result in one import.

cpsencode will also look for filenames given after a leading option flag. Thus, files that occur in options such as **–X***file*, where **X** is any character, will also be examined and, if found, embedded in the output.

EXAMPLES

Let us say you have a cshell script called Figure_8.csh which creates the *PostScript* file Figure_8.ps. Figure_8.csh operates on several data files in order to make the plot. You turn this plot file into a Complete *PostScript* (CPS) file with the command

cpsencode –v Figure_8.csh >> Figure_8.ps

Alternatively, let that be the last command in the script so that it is automatically done by the script itself.
To unscramble the CPS file, simply say

cpsdecode –v Figure_8.ps

BUGS

cpsencode assumes good script etiquette so that any temporary files created by your script are removed before **cpsencode** is called. Otherwise they will take up unnecessary space in the Complete *PostScript* (CPS) file. **cpsencode** will have trouble if you script changes directories with the *cd* command since filenames become relative to another directory. Using absolute paths name for files (starting with /, ˜/. or ˜user/) is bad practice since other users may not be able to access those files.

AUTHOR

Paul Wessel, Geology & Geophysics, SOEST, University of Hawaii, 1680 East-West Road, Honolulu HI 96822, USA. www.soest.hawaii.edu/wessel.