



# DECUS

## PROGRAM LIBRARY

DECUS NO.	8-649
TITLE	QPIP - OS/8 DIRECTORY EDITING PROGRAM
AUTHOR	B. D. Monahan
COMPANY	Standard Telecommunication Laboratories, Ltd. Harlow, Essex, England
DATE	August 28, 1973
SOURCE LANGUAGE	PAL-8

### ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.



## QPIP - OS/8 DIRECTORY EDITING PROGRAM

DECUS Program Library Write-up

DECUS NO. 8-649

This programme provides a few file management facilities not available with OS/8 P1P. These are:

1. Deleting files undeletable by P1P.
2. Reserving space on a directory device.
3. Changing names of files without having to transfer their contents.
4. Handling inquiries for device information.

To do this the programme incorporates its own command decoder, with syntax identical to the usual one, but allowing any characters except "@" in device names, file names and extensions.

## QP1P Write Up

### Objective:

Having used OS/8 for a year, it became obvious that several file management facilities were occasionally desirable, but were not conveniently available. The programme SP1P by D.M. Kristol included on the dectape of DECUS8 - 497 fills many of these gaps, and this is an attempt to fill a few more. The following is a list of some of the problems solved by QP1P.

1. Fortran, Basic and machine code programmes, and the Basic editor, if used carelessly can produce files whose names contain non-alphanumeric characters. These files can be neither moved nor deleted with P1P, and cannot be accessed by other conventional system programmes.

2. It very often happens that a wanted file is deleted, perhaps by creating a new file with the same name, and in such cases one can often see the empty space left by the deleted file in a directory listing. Normally all that is required to recover the file is to put a name to this space.

3. Many programmes create files with temporary names - e.g. BASIC.WS, PAL8BN.TM, etc. which a user may decide later are worth preserving the only way to do this previously was to copy the file to a new name with P1P. This is no trouble on an empty disk, but can be very tedious with full dectapes.

4. When sorting files on a mass storage device (e.g. for maximum overlaying speed) it is easier to rearrange them if space can be reserved to put files in later.

5. An inquiry for device information can clarify puzzling situations. Two of our users spent a very worried morning blaming hardware faults having inadvertently assigned "DTAØ" to dectape unit 1 and "DTA1" to dectape unit Ø.

## Programme Description

To satisfy the requirement of accessing files whose names contain non-alphanumeric characters, the programme contains its own command decoder. This is completely compatible with the normal OS/8 command decoder on valid input, and produces a similar table from the file specifications and options. Its use should cause no problems to an experienced OS/8 user, but since it may be extracted for use in other programmes it is described more fully. It consists of two independent subroutines, a line input subroutine and a line analysis subroutine.

The line input subroutine inputs characters from the keyboard and places them unpacked in a buffer. A zero word marks the end of text in this buffer. The subroutine returns with a pointer to the first character of the buffer in the AC. All the normal editing facilities are available. The subroutine exits on either alt-mode, carriage return, or buffer overflow (72 characters). Carriage return or alt-mode are not put into the buffer, but a page 0 flag is set. The buffer size may be changed by altering the parameter in the Z BLOCK pseudo op. just after the label B0. There are minor differences in response as compared to the normal command decoder.

1. A new "\*" is printed after cont./U. or line feed.
2. Illegal non printing characters are completely ignored and do not echo.
3. If one is between rubout back slashes and cont./U. or cont./C. is typed the rubout brackets are closed before the ↑U or ↑C echo.

The line analysis subroutine is entered with a pointer to the start of the line in the AC. If it is successful it packs the analysed line in a standard format into the buffer at B1 and returns with a skip. Otherwise it prints an error message and does not skip. The buffer at B1 is divided into two parts, one for input files and the other for output files. The numbers of each of these are currently set to 3 of each, but may be changed in the source by altering the Z BLOCK arguments. The line analysis routine does not look up input file names, the square bracket option can be used on both input and output file names and can specify a number up to 4095. Consequently all file specifications occupy 6 words as follows:-

1. Device number (0 if no device specified).
2. Square brackets option (0 if not used).
- 3 to 6 Packed 6 bit file name and extension in the usual format.

All normal command decoder options are handled, with some extensions as follows:-

1. Since the "=" option is not packed with the alt-mode switch, numbers to 7777 7777 are permitted.
2. The square brackets option may be used with input as well as output files, may specify up to 4095, and may be embedded within device name, file name or extension.
3. Except for the following characters, all printing characters including space may be used in device names, file names and extensions. The exceptions are "." ", "<" "←" ":" "@" "=" "/" "(" "[".

All of these characters except "@" may be used by protecting between two "@" characters. The first "@" character disables checking of special characters while the second re-enables it. Apart from this "@" characters are ignored.

All characters are packed correctly, unlike the packing of "\*" by the normal command decoder in special mode.

4. Some forms of invalid syntax accepted by the normal command decoder are rejected by this one.

The main programme calls the two subroutines comprising the command decoder to obtain a specification of its operation. The option switches R, I, C are checked in turn, and the first one found set is acted upon. The others are ignored. Afterwards the P option is checked.

The R option uses the USR to look up an input file and having found it, USR pointers are used to modify the directory segment in core and rewrite it. Each input file is looked up in turn.

The I option reads from device tables in the USR and the top of field 1 the information about each input device and puts this in a readable ASCII form to the first output file.

The C option takes each output file in turn. If the specified size (square brackets option) is non zero a file of that name is opened using the USR. The USR is then called to close the file with the specified size even if this was zero.

The P option uses the USR to chain to P1P.SV on the system device.

### Loading and Saving Instructions

Assuming that the paper tape of the PAL8 source file has been transferred to DSK:QP1P.PA, the following sequence of instructions will produce a working version:

```
.R PAL8
*QP1P < QP1P
.R ABSLDR
*QP1P/9§                               (alt-mode typed)
.SA SYS:QP1P
```

The programme may then be run with the instruction:

```
.R QP1P
```

## INSTRUCTIONS FOR QPIP

THE PROGRAMME IS A GENERAL OS/8 DIRECTORY MODIFYING PROGRAMME

FACILITIES ARE

- 1) RENAMING FILES
- 2) CREATING FILES OF SPECIFIED NAME AND SIZE
- 3) DELETING FILES
- 4) PRINT INFORMATION FROM SYSTEM DEVICE TABLES

WHEN THE PROGRAMME IS RUN IT PRINTS AN "\*" ,AND ACCEPTS A LINE OF COMMANDS IN A SIMILAR MANNER TO THE NORMAL COMMAND DECODER. HOWEVER A MUCH WIDER RANGE OF CHARACTERS IS AVAILABLE. ALL PRINTING CHARACTERS INCLUDING " ", BUT EXCLUDING "\*" MAY BE USED IN DEVICE NAMES, FILE NAMES AND EXTENSIONS. MOST CHARACTERS MAY BE USED DIRECTLY, BUT THOSE HAVING SPECIAL MEANING TO THE COMMAND DECODER (E.G. "<") MUST BE PROTECTED BY A PAIR OF "@" CHARACTERS.

APART FROM THIS THE COMMAND LINE IS CONSTRUCTED AS USUAL. 3 INPUT, AND 3 OUTPUT FILES ARE PERMITTED. THE PROGRAMME USES OPTIONS "C","R","I","P". THERE ARE NO DEFAULT EXTENSIONS.

THE "C" OPTION CREATES FILES AS SPECIFIED IN THE OUTPUT FILE LIST , DELETING OLD FILES OF THE SAME NAMES. IF THE FILE SIZE IS 0 THEN A NEW FILE IS NOT CREATED. THIS MAY BE USED TO DELETE FILES UNDELETABLE WITH PIP BECAUSE OF THEIR NAMES OR TO RECOVER FILES ACCIDENTALLY DELETED. DEFAULT DEVICE IS DSK.

THE "R" OPTION RENAMES FILES IN THE DIRECTORY. EACH INPUT FILE HAS ITS NAME CHANGED TO THE CORRESPONDING OUTPUT FILE NAME. THIS MAY BE USED WHEN A PROGRAMME INCORRECTLY PRODUCES MULTIPLE FILES OF THE SAME NAME, OR FILES WITH NAMES CONTAINING INVALID CHARACTERS, OR EVEN TO MAKE SOME FILES ON A SYSTEM UNAVAILABLE TO AN ORDINARY USER. DEFAULT INPUT DEVICE IS DSK. DEFAULT OUTPUT DEVICE IS SAME AS CORRESPONDING INPUT DEVICE AND THE COMMAND IS INVALID IF THE DEVICES ARE DIFFERENT.

THE "I" OPTION LOOKS UP DEVICE INFORMATION IN THE SYSTEM TABLES FOR EACH INPUT DEVICE, AND OUTPUTS A SUMMARY TO THE FIRST OUTPUT FILE. OTHER OUTPUT FILES ARE IGNORED. THE DEFAULT OUTPUT DEVICE IS DSK AND THERE ARE NO DEFAULT INPUT DEVICES.

THE "P" OPTION CHAINS TO PIP ON COMPLETION.



EXAMPLE OF USE

```
.R PIP
*DTA2:/E

  9/3/73
<EMPTY> 730
  730 FREE BLOCKS

*!C
```

START WITH A CLEAN DECTAPE

```
.R OPIP
*DTA2:TEM1.XX[17],DTA2:TEM2.YY[300]</C
*/P
*DTA2:/E

  9/3/73
TEM1 .XX 17 9/3/73
TEM2 .YY 300 9/3/73
<EMPTY> 413
  413 FREE BLOCKS

*!C
```

START OPIP  
CREATE 2 FILES AS SPECIFIED  
CHAIN TO PIP

PIP LISTS THE NEW DIRECTORY  
NOTE THAT SIZES GREATER  
THAN 256 BLOCKS MAY BE  
SPECIFIED

```
.R OPIP
*TEM3!U
*DTA2:TEM3.ZZ[1],DTA2:TEM4.&&[3]</C/P
*DTA2:/E

  9/3/73
TEM1 .XX 17 9/3/73
TEM2 .YY 300 9/3/73
TEM3 .ZZ 1 9/3/73
TEM4 .&& 3 9/3/73
<EMPTY> 409
  409 FREE BLOCKS

*TEM2.YY</D!U
DTA2:TEM2.YY</D
*DTA2:/E

  9/3/73
TEM1 .XX 17 9/3/73
<EMPTY> 300
TEM3 .ZZ 1 9/3/73
TEM4 .&& 3 9/3/73
<EMPTY> 409
  709 FREE BLOCKS

*!C
```

INSTRUCTION ABORTED BY CONT/U  
CREATE 2 MORE FILES AND CHAIN TO  
PIP ON COMPLETION

NOTE THAT NON ALPHANUMERIC  
CHARACTERS ARE PERMITTED

ORDINARY USE OF PIP

```
.R QP IP $*TEM2.\.2MET\DTA2:TEM2.BB[300]</C
*TEM4.tU
*DTA2:TEM4.AA<DTA2:TEM4.&&(RP)
*DTA2:/E
```

CREATE 1 FILE (NOTE RUBOUT)  
COMMAND ABORTED  
CHANGE A NAME AND THEN CHAIN TO PP

```
9/3/73
TEM1 .XX 17 9/3/73
<EMPTY> 300
TEM3 .ZZ 1 9/3/73
TEM4 .AA 3 9/3/73
TEM2 .BB 300 9/3/73
<EMPTY> 109
409 FREE BLOCKS
```

PIP For files longer than 255 blocks the VSR cannot allocate the smallest space, so TEM2.BB does not go back where it was.

```
*TEM4.AA</D
ERROR DELETING FILE
*DTA2:TEM4.AA</D
*!C
```

```
.R QP IP
*TEM5.tU
*DTA2:TEM5.AA[3],TEM1.\.1MET\DTA2:TEM1.XX</C/P
*DTA2:/E
```

RUBOUT

CREATE 1 FILE AND  
DELETE ANOTHER - CHAIN TO PIP

```
9/3/73
<EMPTY> 317
TEM3 .ZZ 1 9/3/73
TEM5 .AA 3 9/3/73
TEM2 .BB 300 9/3/73
<EMPTY> 109
426 FREE BLOCKS
```

PIP Smaller files are more easily recovered, and TEM5.AA contains the same data that TEM4.AA did before it was deleted above by PIP

```
*!C
```

.R QPIP  
\*DTA2:@...<?@.A@(@<TEM3.\.3MET\DTA2:TEM3.ZZ/R  
\*TTY:<DTA2:.,DSK:/I

RUBOUT

QPIP DEVICE ENQUIRY BDM JUL.73

DEVICE INFORMATION CURRENTLY IN SYSTEM TABLES

DEVICE NO. 1  
SYSTEM DEVICE NO. 0012  
HANDLER SAVED ON BLOCK 0017  
ENTRY AT 0012  
DEVICE CONTROL WORD 4160

DEVICE NO. 2  
SYSTEM DEVICE NO. 0002  
PERMANENTLY RESIDENT HANDLER  
ENTRY AT 7607  
DEVICE CONTROL WORD 4060

CHANGE A NAME.  
NOTE USE OF @ TO PROTECT  
SPECIAL CHARACTERS

INQUIRE ABOUT DTA2  
AND DSK.  
SINCE NO ODD ASSIGNMENTS  
ARE IN FORCE, RESULTS ARE  
JUST AS EXPECTED

\*(P)  
\*DTA2:/E

9/3/73  
<EMPTY> 317  
..<? .AC 1 9/3/73  
TEM5 .AA 3 9/3/73  
TEM2 .BB 300 9/3/73  
<EMPTY> 109  
426 FREE BLOCKS

\*DTA2:</Z  
\*+C

CHAIN TO PIP

THIS SHOWS THE RESULTS OF THE  
RENAMING ABOVE.  
NOTE THAT NO CONVENTIONAL  
PROGRAMME CAN ACCESS THE 1 BLOCK  
FILE, THOUGH IT IS STILL MOVED  
IN A SQUISH

