

POLYCORP

The material presented in this document has been expressly prepared by Polycorp New Zealand Limited.

No part of this publication may be reproduced, stored in a retrieval system, transmitted in any form or by any means, electronics, mechanical, photocopying, recording or otherwise without prior permission of Polycorp New Zealand Limited.

Copyright © 1982
Polycorp New Zealand Limited

CONTENTS

	PAGE
1. INTRODUCTION	1
1.1. WRITING A PASCAL PROGRAM	2
1.2. TRYING OUT PASCAL	3
1.3. CHANGES MADE TO OMEGASOFT PASCAL	4
1.4. OMEGASOFT PASCAL	5
1.5. EXTENDED VERSION OF PASCAL	5
1.6. OMEGASOFT ASSEMBLER	5
1.7. USE OF COLOUR	6
1.8. ASCII Control Codes	7

INTRODUCTION

The Omegasoft Pascal language is supplied in a form so that it runs with the minimum of user interaction. Those users with an extensive knowledge of computing can request an extended version which allow the user more flexibility in how they run and use Pascal.

Omegasoft Pascal programs are written using the text editor resident on the Poly. The programs are then SAVE'd onto disk and the Pascal compiler loaded.

The compile has three phases

1. Compilation of the Pascal source code into assembler.
2. Assembling of the code in 6809 machine language.
3. Linking in of the subroutines used in the program.

The three phases are run with a DOS command with the minimum of options. The extended Pascal allows the individual phases to be run separately, but requires the user to understand a considerable amount about the Poly. It is not recommended that the extended version be used until considerable experience has been gained using the standard options.

The files supplied for the standard Pascal are:

PASCAL.CMD	A DOS command which sets the standard options and runs the compile.
PC.CMD	The Pascal compiler.
RA.CMD	The relocatable assembler.
LL.CMD	The linking loader.
PASLNK.R0	A linking file.
RE.CMD	The reenter command.
RL.R0	The subroutine library.
PRIMES.TXT	A program written in PASCAL.

These files take approximately 320 disk sectors and must be on the disk being used when a compile is done.

Following the compile, the compiled program can be run as a DOS command.

1.1. WRITING A PASCAL PROGRAM

The Poly text editor is described in the Utilities Manual. The steps to use it are:

1. Load the text editor.
This is done either by typing in
TEXT <ENTER>
from either DOS or BASIC
Ready
is displayed in CYAN.
2. Type in the PASCAL program.
The editing commands are described fully in the Utilities Manual.
Note that the line numbers shown on the screen are for editing purposes only and are removed when the program is SAVE'd onto disk.
3. SAVE the program onto disk.
Use the SAVE command. This saves the program as filename.TXT.
For example:
SAVE "MYPROG"
saves your program as MYPROG.TXT.
4. Run the Pascal compiler.

This is run by using a DOS command in the form
PASCAL filename [S] [D] [L] <ENTER>

The filename is the name of the source file previously saved on disk. This is compiled into a file with name filename.CMD.

If S is specified then the compile only checks the program for syntax errors. If S is not specified and errors are found during the compile, the compile is terminated.

If L is specified, then a listing of the compilation on the printer is made.

If D is specified as well as L then the listing is displayed on the screen.

For example:

DOS <ENTER>

PASCAL MYPROG SL <ENTER>

Compiles the source file MYPROG.TXT for syntax only. A listing is produced on the printer.

PASCAL MYPROG.TXT LD <ENTER>

Compiles the source file MYPROG.TXT , and if no errors are found, then the compiled file MYPROG.COM is created. When the compile is terminated the Poly is left in DOS mode. To run the compiled program simply type in filename <ENTER>

For example:

MYPROG <ENTER>

Warning: Do not run Pascal programs from either BASIC or TEXT using the form +filename. If in either BASIC or TEXT mode, enter DOS first by typing in DOS.

1.2. TRYING OUT PASCAL

The following Pascal program is supplied on disk:

```
program fastprimes(INPUT,OUTPUT);
(* find the first 1229 primes *)
const n=1229; nl=35; (* sqrt of n *)
var i,k,x,inc,lim,square,lin: integer;
    prim: boolean;
    p,v: array(.0..n.) of integer;
begin
    writeln;
    write(2:8,3:8); lin:=2;
    x:=1; inc:=4; lim:=1; square:=9;
    for i:=3 to n do
    begin (* find next prime *)
        repeat x:=x+inc; inc:=6-inc;
            if square<=x then
            begin lim:=lim+1;
                v(.lim.):=square; square:=p(.lim+1.)*p(.lim+1.)
            end;
            k:=2; prim:=true;
            while prim and (k<lim) do
            begin k:=k+1;
                if v(.k.)<x then v(.k.):=v(.k.)+2*p(.k.);
                    prim:=x<>v(.k.)
            end
        until prim;
        if i<=nl then p(.i.):=x;
        write (x:8); lin:=lin+1;
        if lin=9 then
            begin writeln; lin:=0
        end
    end;
    writeln
end.
```

It is held in the file PRIMES.TXT
Enter the TEXT editor and

LOAD "PRIMES <ENTER>
When complete

LIST <ENTER>
and the file is displayed on the Poly with line numbers.
Move the cursor up onto one of the lines and make a minor
change and press <ENTER>.

Type in

SAVE "NEWPRIME
and the altered file is saved as NEWPRIME.TXT.

Type in

+PASCAL NEWPRIME L <ENTER>

When complete
DOS
is displayed.

Type in

NEWPRIME <ENTER>
to run the program.

Shift <EXIT> may be pressed at anytime to terminate the program
and return the Poly to DOS.

To reedit the program type in

TEXT <ENTER>

and when Ready is displayed type in

LOAD "NEWPRIME

1.3. CHANGES MADE TO OMEGASOFT PASCAL

The device AUXOUT documented in the manual as the printer, has
not been implemented. To print on the printer, open a text file
with .PRT as the extension. When the file is closed, it is
printed and deleted.

The Poly does not have the square or curly brackets in the
character set. The last two pages of the OMEGASOFT PASCAL Manual
show how (* and *) are used for remarks and (. and .) for arrays.
Care must be taken when using .) to leave a space before the .)
when the last element of the subscript is numeric.

For Example:

SQUARE := V(.5.)
will cause a compilation error.

This should be written

SQUARE := V(.5 .)

If a FLEX system is available this can still be written

SQUARE := V[5]

1.4. OMEGASOFT PASCAL

Pascal comprises:

- 8 files listed earlier in this manual.
- Introduction to Omegasoft Pascal on Poly (this manual).
- Omegasoft Version 2 - Pascal Language Handbook.

1.5. EXTENDED VERSION OF PASCAL

The extended version of Pascal provides in addition to the above:

- Pascal Utilities Manual.
- Pascal Configuration Manual for POLY.
- How to use the Linkage Creator.

as well as further utilities on disk.

1.6. OMEGASOFT ASSEMBLER

The Omegasoft assembler comprises:

The files RA.CMD	The Relocatable Assembler
LL.CMD	The Linking Loader

The Omegasoft Version 1.2 Relocatable Assembler and Linking Loader manual.

1.7. USE OF COLOUR

The screen is displayed using the Teletext conventions but PASCAL programs execute in ASCII mode. In order to use the Teletext control characters, a shift must be made into Teletext mode before using the Teletext control characters, and a shift back to ASCII after using them. The example shows how the CHR function is used to incorporate these in an expression.

The teletext control characters are:

<u>ASCII DECIMAL</u> <u>VALUE</u>	<u>FUNCTION</u>
0	Not used
1	Starts RED characters
2	Starts GREEN characters
3	Starts YELLOW characters
4	Starts BLUE characters
5	Starts MAGENTA characters
6	Starts CYAN characters
7 *	Starts WHITE characters
8	Starts FLASHING
9 *	Ends FLASHING
10	Not used
11	Not used
12 *	Normal height
13	Double height **
14 *	Shift to ASCII
15 *	Shift to Teletext
16	Reverse video on
17	Starts RED graphics
18	Starts GREEN graphics
19	Starts YELLOW graphics
20	Starts BLUE graphics
21	Starts MAGENTA graphics
22	Starts CYAN graphics
23	Starts WHITE graphics
24	CONCEAL display on rest of line
25 *	Contiguous graphics
26	Separated graphics
27 *	Reverse video off
28 *	No background to characters
29	Set background to current colour
30	Print graphics characters over control characters
31 *	Print space for control characters

Each of the control characters take up ONE screen position except the reverse video on and off characters and the ASCII and Teletext shift characters. All control characters are reset at the beginning of each line to those with an * beside them. Reverse video is switched off at the end of each WRITE.

Double height characters extend down to the following line. If double height is used anywhere on a line, the following line is not displayed.

For example:

```
Program CHR_TEST (Input, Output);
Var :
  N : integer;
  C : char ;
Begin
  C := CHR(1)
  A := CHR(14)
  T := CHR(15)
  N := 0 ;
  While (N < 100)
    Write (T,C,A,N)
    N := N+1 ;
  End.
```

This prints out the numbers 1 to 100 in red. Note the shift to and from Teletext before and after the red control code.

ASCII Control Codes

ASCII	Decimal	Value	Function
	7		Beep
	8		Moves the cursor 1 space to the left
	9		Moves the cursor 1 space to the right
	10		Moves cursor down 1 line
	11		Moves cursor up 1 line
	12		Clears screen and moves the cursor to the home position
	13		Moves cursor to the start of the line (RETURN)
	14		Shift to ASCII mode
	15		Shift to Teletext mode
	16		Reverse video on
	27		Reverse video off
	30		Clear to the end of the line.