

**WATCOM**

Waterloo Computing  
Systems Limited

158 University Ave. W.,  
Waterloo, Ontario  
Canada N2L 3E9

(519) 884-2700

# InfoWAT

Volume 1, Issue 5

February 1983

## *VAX/VMS Software*

### **Overview**

The Waterloo interactive language interpreters for APL, BASIC, COBOL, FORTRAN and Pascal have now been implemented on the Digital VAX computer with the VMS operating system. A general text editor has also been implemented. The software operates in full-screen mode when used with DEC VT100, GIGI, and VT52-compatible terminals. The GIGI terminal can also be used with the Waterloo APL interpreter to produce APL characters including overstrikes. The software runs in a line-oriented mode when used with other terminals.

The language processors have been designed to produce meaningful, concise and specific error diagnostics. They also include interactive debugging facilities. These characteristics make the interpreters useful in either a program development or a teaching environment. On-line reference information for each processor is available using the interactive HELP facilities of the interpreters and VMS.

This software is presently available for Commodore's SuperPET, IBM's 370 with VM/CMS (VM/SP), IBM's Personal Computer with MS-DOS, and the Northern Digital microWAT. It is hoped that this software will soon be implemented on Digital's Professional 350 micro system as well. Since the same software has been implemented on these different systems, application programs developed with one of the interpreters can be used on a variety of computer systems without modification.

### **Full-Screen Editing**

Digital's EDT text editor provides excellent full-screen editing capabilities on VAX/VMS. Although providing a simpler, more modest editing facility, the Waterloo Editor can provide substantially superior real-time response to keystrokes on busy VAX timesharing systems when compared with EDT. In full-screen mode, the Waterloo Editor supports editing of text without typing of commands, in a manner similar to Digital's EDT. In addition, the Waterloo Editor's command set provides features such as global and selective multi-line changes. The numeric keypad and program-function keys are used to control viewing and editing of text. The Waterloo Editor can also be used to edit files containing APL characters.

The Waterloo Editor is the same editor available for the Commodore SuperPET, IBM 370 with VM/CMS, IBM PC with MS-DOS, and the Northern Digital microwAT. Commands used on these systems can be used with the VAX/VMS version of the Waterloo Editor.

## **Interpreter Features**

### **APL**

The Waterloo APL interpreter is intended to be a complete and faithful implementation of the IBM/ACM standard for APL with respect to the syntax and semantics of APL statements, operators and primitive functions, input and output forms, and defined functions. System commands, system variables and system functions are those consistent with a single user environment. There are no significant design limitations on the rank or shape of arrays, the length of names, or the size of workspaces. Sequential- and relative-access files are supported. Use of Digital's RMS file system permits file compatibility with other high-level languages. The shared variable processor is omitted. Extensions include system functions supporting files of APL arrays. APL equivalents of the BASIC features PEEK, POKE and SYS are included.

### **BASIC**

The Waterloo BASIC interpreter includes full ANSI X3.60-1978 BASIC. Numerous extensions include structured programming control, long names for variables and other program entities, powerful character-string manipulation, callable procedures and multi-line functions, sequential and relative file capabilities, integer arithmetic, debugging facilities, and convenient program entry and editing facilities.

### **COBOL**

The Waterloo COBOL interpreter includes a substantial implementation of ANSI X3.23-1974 COBOL. It supports level 1 and parts of level 2 in the following modules: Nucleus, Sequential I-O, Relative I-O and Table-handling. The interpreter provides a powerful interactive debugging facility.

### **FORTRAN**

The Waterloo FORTRAN interpreter supports a special dialect designed for teaching purposes. It has many of the characteristics and much of the flavour of normal FORTRAN, but varies significantly from established standards for that language. This language processor has many of the important characteristics of the WATFIV-S compiler which is widely used on IBM computers, plus some features from the new FORTRAN-77 definition. Examples of language features supported are: FORMAT, subroutines and functions, multi-dimensional arrays, extended character-string manipulation, structured programming control and file input/output. The interpreter also provides a powerful interactive debugging facility.

### **Pascal**

The Waterloo Pascal interpreter is an extensive implementation of Pascal, corresponding very closely to draft proposals being produced by the International Standards Organization (ISO) Pascal committee. The ISO draft language is a refinement of the language originally defined by Wirth, varying only in minor aspects. This implementation includes sophisticated features such as text file support, pointer variables, multi-dimensional arrays, and an extension which facilitates string processing. A significant feature of Waterloo microPascal is its powerful interactive debugging facility.



## Interpreters on a Mainframe?

In interactive environments, interpreters simplify the process of developing and debugging programs. Use of compilers to develop programs requires multiple steps at each attempt: EDIT, COMPILE, LINK, RUN. Interactive interpreters provide a simple, integrated program development environment with superior debugging facilities during program execution. When errors occur during execution, the current source line is displayed, the approximate position of the error in the line is indicated, and a diagnostic message appears.

Interpretive language processors prove to be less expensive, in terms of memory resources, than multi-step program development tools. As well, the interactive nature of interpreters leads to a high level of productivity during program development. These factors can yield superior throughput and resource utilization for student programming usage.

### **BASIC: Digital's versus Waterloo's**

Digital makes available a very good interactive BASIC language processor for the VAX with VMS. When compared with Digital's BASIC, the Waterloo BASIC interpreter has several important advantages. One of the principal advantages is a set of control statements for Structured Programming which are simpler and more general than those provided by Digital's product. File input/output statements in Waterloo's BASIC are also simpler and more general. Digital's BASIC input/output statements provide the means to use special features of VAX devices that the Waterloo interpreter does not provide. Unfortunately, the means of providing this capability adds complexity to the language, making simple file-processing tasks difficult for novice students.

Another significant advantage of Waterloo's BASIC is the ability to run application programs, developed using it, on a variety of computer systems without modification. The interpreter also requires substantially less memory than Digital's compiler.

The Waterloo BASIC interpreter has been used since September 1982, on a test-site basis, for an introductory programming course at the McMaster University School of Business (Hamilton, Ontario). Waterloo's interpreter was found more effective and economical, in their usage, than Digital's BASIC. Digital's BASIC is not a multi-step program development tool, but rather a compiler oriented to interactive use. Each time a program is run using Digital's BASIC, if a change has been made, the entire program is re-compiled. This is not necessary with an interpreter. In introductory programming courses, students repeatedly correct errors and run their programs, until they run successfully once.

It was also observed at McMaster that students required less attention from tutors when using the Waterloo BASIC interpreter than when using Digital's BASIC. This can be explained as the result of simpler control and input/output structures in the language, as well as source-text oriented diagnostic and debugging features.

### **Availability**

The VAX version of this software is available at modest cost from:

WATSOFT Products Inc.  
158 University Avenue West  
Waterloo, Ontario  
Canada N2L 3E9  
telephone (519) 886-3700  
telex 06-955458

## *Index Expressions in APL*

An error in the evaluation of the individual components of an index expression can cause errors or incorrect results when using Waterloo microAPL with a fragmented workspace. The problem exists only with Version 1.1 on the Commodore SuperPET and the IBM Personal Computer. The index expression evaluates incorrectly when a memory compaction (garbage collection) occurs during the evaluation. The likelihood of this occurring can be reduced by forcing a memory compaction (e.g. `M ← □WA`) from time to time. The problem can be avoided by evaluating the individual expressions, assigning the results to variables, and then performing the index operation. For example,

```
X ← A[ 1+1;2+2 ]
```

changes to

```
A1 ← 1+1
```

```
A2 ← 2+2
```

```
X ← A[ A1;A2 ]
```

## *Simulating a GET in Pascal*

The program below simulates a BASIC GET statement to read a character from the keyboard.

```
program main( input,output );
var
  c : char;
  io : file of char;
begin
  writeln( 'Character?' );
  reset( io,'keyboard' );
  c := chr( 0 );
  while( c = chr( 0 ) )do
    read( io,c );
  writeln( 'Character="',
          c,'" );
end.
```

Until a key is hit, the read statement returns a null value in the variable c. When a non-null value is returned, the while loop is terminated and the character is displayed by the second writeln statement. The character entered is not echoed directly on the screen when it is typed.

## *Form Feeds and SuperPET Printer Output*

Normally a SuperPET printer is dedicated to the SuperPET to which it is connected. It is up to the person using it to ensure that the printer is positioned to the start of a page before printing a listing. If the printer is connected to several SuperPETs through a device such as a MUPET, it can be shared by more than one person. It can even be physically distant from one or more of the microcomputers to which it is connected, and adjusting the printer before producing a listing may be difficult or even impossible. In response to requests from SuperPET users who wish to share printers, the following program has been written.

```
010 open #2,"ffpatch,prg",output
020 loop
030   read i
040   if i = 999 then quit
050   print #2, chr$(i);
060 endloop
070 close #2
080
090 data 5,192,0,41,0,0
100 data 134,8,183,5,149,15,50
110 data 57,5,202,174,98,230,2
120 data 193,130,38,22,204,0
130 data 96,52,6,236,100,189
140 data 192,114,204,0,12,237
150 data 228,236,100,189,211
160 data 123,50,98,57,0,0,0,0
170 data 2,0,999
```

The above program is run only once. It creates a file called "ffpatch" which is written to a diskette. The patch program is run from the SuperPET's main menu by entering "disk.ffpatch" when the SuperPET is turned on, or whenever a switch from 6502 mode to 6809 mode is made. When it has completed, the program returns to the menu. The program applies a patch to the system routine called I3ECLOSE\_ so that a FORMFEED character is printed whenever a printer file is closed. This will prevent the output from several users from appearing on one page.