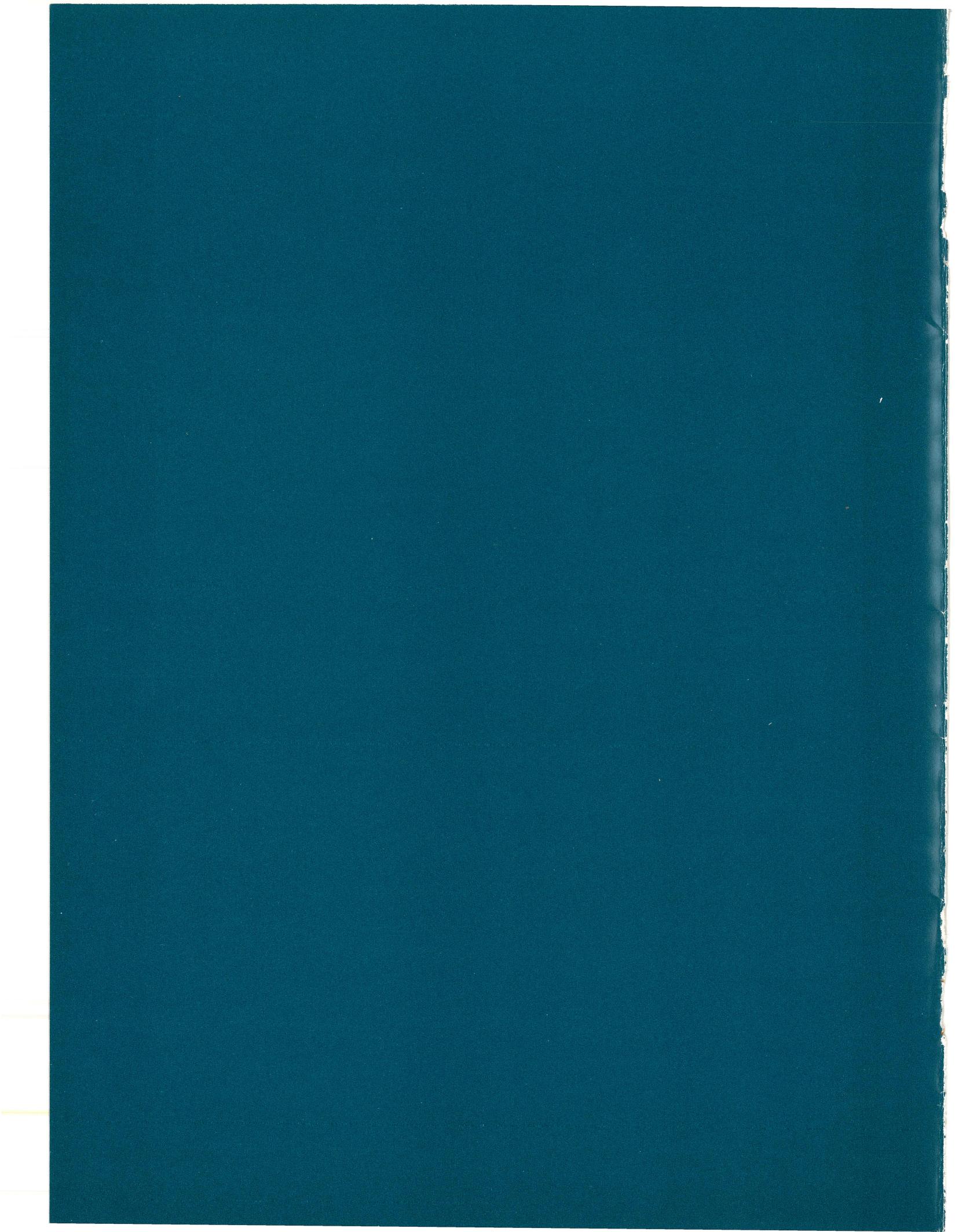




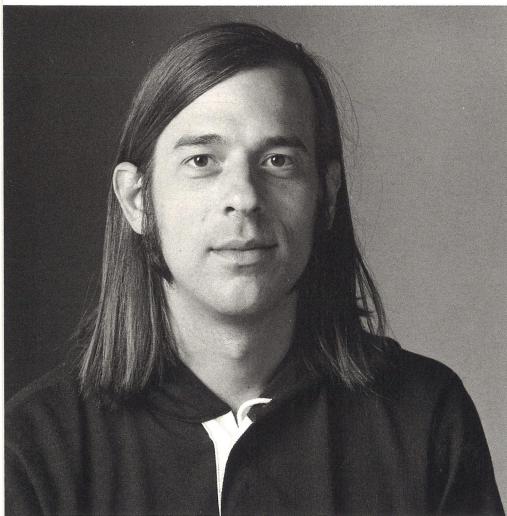
We put a lot of ourselves into Pascal-2



## Pascal-2 on VAX

The creators of Pascal-2 for the VAX/VMS believe strongly in the classical disciplines of Pascal, and have built that integrity into all of Oregon Software's products. Pascal-2 is the only language compiler that provides uniform software development tools for the full line of Digital Equipment Corporation computers from the Professional 350 to the VAX. Here is their own testimony about the development of Pascal-2 on VAX.

# Pascal and Pascal-2



**Don Baccus**  
*President, Oregon Software*  
Pascal-2 Co-Author and  
Code Generation Designer

“Natural expressiveness in a programming language lets you write code that closely fits the way you think about programming problems. If a language allows you to express yourself to the machine in something that corresponds closely to how you think about the program, then it becomes relatively easy to do the programming.

The person who created Pascal looked at the kinds of problems that programmers solve and tried to identify the most natural way of expressing the problems. He built the language around those ideas.

As a result, there is a very close correlation between Pascal and the way programmers solve problems. That is Pascal’s greatest strength. It provides you with a programming methodology that helps you write a reliable, efficient and readable program.

**Reliability** The structure of Pascal demands a great deal of discipline from a programmer, but that prevents a lot of mistakes. The resulting programs are not only efficient, they are also very readable because the form and structure are so expressive and so carefully defined.

**Readability** People have to read programs. In order to maintain or expand a program, somebody’s got to read it. And it’s usually not the person who wrote it in the first place.

If a programmer has to spend hours deciphering something that is unreadable, you’re going to pay for those hours. The readability of the original program has a direct impact on the cost of maintenance.

Pascal-2 has been so **Pascal-2** successful in the last few years because it has earned a reputation as a well-constructed compiler that produces efficient, readable programs. It does just what a compiler should do — takes care of all the low-level work and leaves programmers free for creative problem solving.”

“The VAX is the most popular software development environment available today. It's a working computer that is being heavily used by system developers, scientists, engineers, programmers, and academicians.

**Flexible Power** Programming requirements for these groups are often very different, but they're all equally demanding. Systems developers need machine access. Scientists and engineers want fast execution speed and use of floating point numbers. Students need good, descriptive error messages. In order to make Pascal-2 a good language development tool for all of these people, we had to make it flexible as well as powerful.

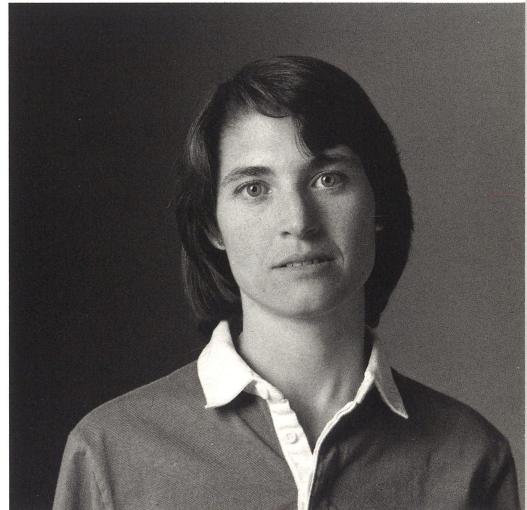
We all **Dependability** want dependability in our development tools. At Oregon Software we use our own tools as much as possible. For example, Pascal-2 was used in the development of the library and utilities; the debugger was used to help track down bugs in the compiler, and the profiler to track overall performance. The basic design of these products has not changed much in the last few years. We use them. They are sound products.

**Effective Tools** I have a real appreciation for effective tools. I've been in situations where I've had to use ‘writeln’ statements to debug a program, count statements, and use formulas to determine where the ‘bottleneck’ was, or wade through listings to find a procedure and spend hours in an editor trying to get my program to look nice and neat.

With the Pascal-2 utilities available to me, my life is so much simpler. Using the debugger, I can get the information I need by ‘breaking’ in the middle of my program. The profiler output lets me find bottlenecks at a glance. The XREF output allows to me to quickly determine where a procedure is defined or used.

Running a source file through PASMAT only takes a few minutes and I end up with a perfectly formatted Pascal file.

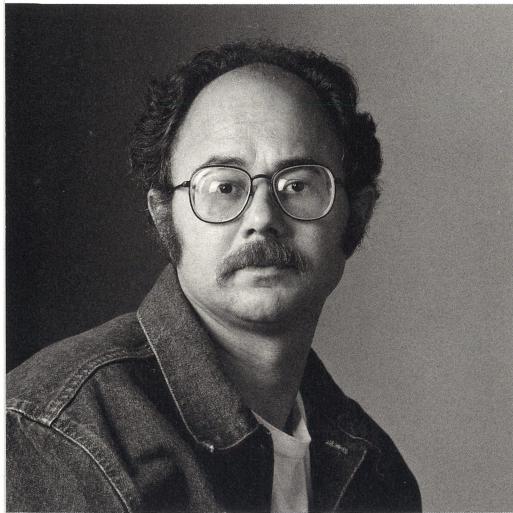
Generally, I think VAX programmers will be able to solve any programming problem with Pascal-2. It's an effective set of tools for day-to-day use.”



**Lynn Conway**  
*Software Engineer*

Designer of Runtime Support Routines  
and Ported Debugger

# Portability



**Mike Cavalle**  
*Senior Software Engineer  
Compiler Writer*

“Now that we’re releasing Pascal-2 for the VAX, we have complete portability across the Digital product line from the Pro 350 through the VAX. That’s a significant breakthrough.

Using Pascal-2, a developer writes an application once, no matter what system it’s going to run on. Once the algorithm has been implemented, it can be moved to other DEC machines very quickly. Pascal-2 for RSX is a good example. It’s 50K of Pascal-2 code that was moved to the VMS operating system in a few days.

Other languages, and specifically, other versions of Pascal are implemented in different ways on different DEC machines. Incompatibilities like these can give a developer nightmares. Little ‘system differences’ lead to weeks of code changes. Sometimes complete code rewrites are preferable.

Pascal-2 is implemented the same way on every system, both for DEC and non-DEC machines. Every implementation has identical language features, and identical interface to the operating system and to the hardware.

Pascal-2 is **VAX/VMS** also identical to our implementation on a whole range of VERSAdos and UNIX-based 68000 systems such as V7, System III, System V, and Berkely 4.2. This is a natural migration path for VAX developers.

This makes algorithms expressed in Pascal-2 ‘actually’ portable. It opens up a lot of possibilities, especially to VAX-based developers. VAX users are inventive. They are already using the VAX in such a variety of ways, it will be interesting to see all the new applications they’re going to come up with.”

# Performance

## Pascal Availability

PASCAL-2 IS AVAILABLE ON THE DIGITAL HARDWARE SHOWN BELOW.\*

DEC Hardware (host)	Oregon Software
Pro 350/RT-11	Pascal-2
PDP-11/RSX	Pascal-2
PDP-11/RSTS	Pascal-2
PDP-11/RT-11	Pascal-2
PDP-11/UNIX	Pascal-2
VAX/VMS	Pascal-2
MicroVAX	Pascal-2

\*Pascal-2 is also available for cross-development to the 68000.

Pascal-2's user interface is identical across the DEC line. Regardless of the operating system, the developer uses the same compiler commands, the same switches, receives the same error messages in the same format. The Debugger, Profiler, and other utilities function in the same way. The only differences are minor differences in the command-line syntax; we match the syntax used by operating system, which varies slightly among DEC systems.

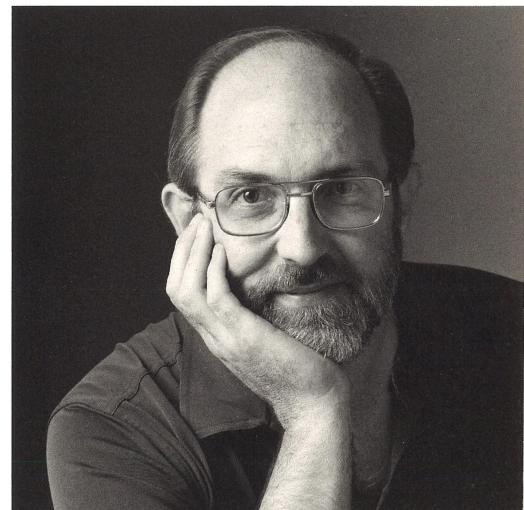
"When I write in assembly language, I strive for the most efficient code possible. It's a matter of principle. I like quality. And quality in a compiler's output means tight code. Pascal-2 does exactly what I'd expect a first-rate optimizing Pascal compiler to do — it generates really efficient code.

In situations where I used to switch to an assembler for the efficiency I wanted, I now just write it in Pascal. I could get the code a little bit tighter 'by hand,' but the small increase in efficiency just isn't worth the bother.

**Quality** I can't stand wasted effort. I'd much rather write a program well the first time by saying exactly what I mean. Pascal-2 allows me to do just that.

When I come back to a program that I've set aside for a while, I can easily recognize what each part of the program does, and how it does it.

Pascal-2 promotes good programming techniques, and contributes to the clear expression of ideas. It's a professional's tool."



**Rick Lahrson**  
*Senior Software Engineer  
Code Generator Author*

# Robust Pascal-2

“Pascal-2 is robust because it’s a solidly built compiler that’s had a chance to age gracefully. In the three years since its commercial introduction, Pascal-2 has been subjected to a lot of rigorous use. It’s been repeatedly tested in a wide variety of industrial, scientific, business and educational applications.

The same basic product has been implemented on several different processors and nearly a dozen operating systems, with sizes ranging from small, single-user RT-11 systems to major VAX/VMS installations.

This compiler is not easy to ‘break’.

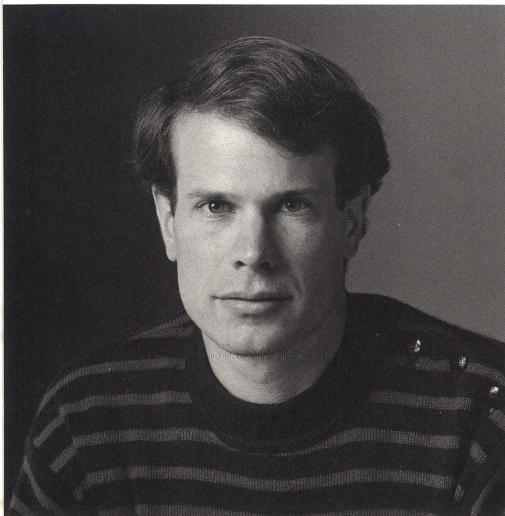
**Design Goals**  
Decisions about design priorities have to be based on established goals. Because the VAX implementation was the final step in a long development process, most of those design goals have been in place for a long time.

For example, Pascal-2 was designed from the outset to produce excellent code. As a result, when writing the VAX code generator, we were able to quickly generate good code as well as emphasize reliability and dependability. And we did this all within the design framework of tuning Pascal-2 for best performance on the VAX.

## Straightforward Answers

At Oregon Software, we seek simplicity and uniformity. This doesn’t rule out complex solutions to complex problems, but achieving straightforward answers is primarily a matter of understanding our goals and keeping these goals clearly in mind.”

**Will Neuhauser**  
*Senior Software Manager*  
Project Leader for VAX/VMS



# Documentation and Support

“Good documentation has to get users in and out of the system as quickly as possible.

After that first step, their needs become more complicated. Our audience includes users with widely different backgrounds. They range from students just learning Pascal to long-time applications programmers who are very demanding.

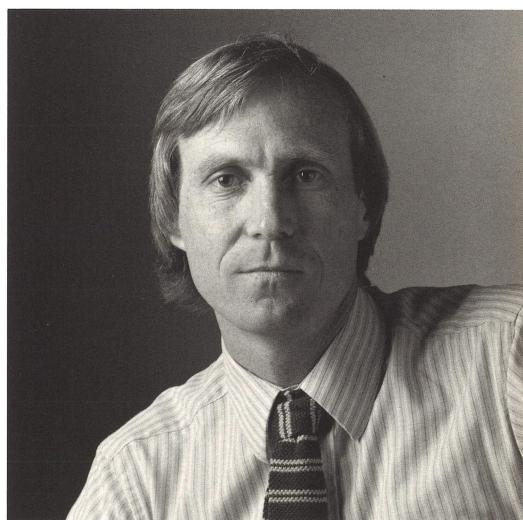
**Detail** The most valuable tool we offer them all is an extensive index. The index can take readers to detailed explanations or functional references with a minimum of delay.

When they're stuck, they want to find a solution quickly, without stumbling through overly difficult sections.

Our style **Organization** is very systematic: the inherent structure of the material dictates the organization of the documentation. The manual is easy to use because the style is consistent and flows logically. Our goal is to explain the material clearly and directly.

**Response** Documentation is only part of our support package. We give quick written or telephone response to a customer's problem. We also supply software updates twice a year as part of the support fee.

We try to provide all the information users will need to fully understand the product and put it to work on their problems.”



**Tom Hanrahan**  
*Technical Writer, VAX Project*

# Pascal-2 from Oregon Software

For the past seven years, Oregon Software has committed its efforts to meeting the needs of professional programmers. We pride ourselves in the development of superior products, documentation and support.

If you would like to put Pascal-2 to work on your VAX, call us at 1-800-874-8501.

## Pascal-2 Specifications

The Pascal-2 system is available on Digital's VAX, MicroVAX, PDP-11/RSX, RSTS, UNIX, TSX-Plus, and Professional Computers running P/OS, RT-11.

The current release of Pascal-2 for VAX/VMS conforms to level 1 of the Pascal standard published by the International Standards Organizations (ISO 7185).

## Features:

- Conforms to the international Pascal standard (ISO 7185).
- Includes conformant array parameters for "flexible" or open arrays.
- Supports all features of standard Pascal, including set types of up to 256 elements and packed data structures.
- Supports integer types of 32 bits.
- Supports single- and double-precision arithmetic.
- Allows calls to separately compiled routines written in Pascal, FORTRAN, MACRO, and all other languages on VMS using DEC's standard calling sequence, providing an interface with existing software.
- Offers "include" directive to combine multiple source files into one compilation.
- Allows flexible declaration order, structured constants, and a default, "otherwise" in the "case" statement.
- Provides I/O extensions to associate external file names or I/O options with Pascal files; to make use of random-access files; to rename or delete files from within a Pascal program.
- I/O Error Trapping allows you to recover from normally fatal I/O errors with your own code.
- Includes a complete set of programmer utilities, including program and text formatters; cross-references for program identifiers and procedures; a dynamic string package written in standard Pascal; and a set of definitions to allow Pascal-like use of MACRO.

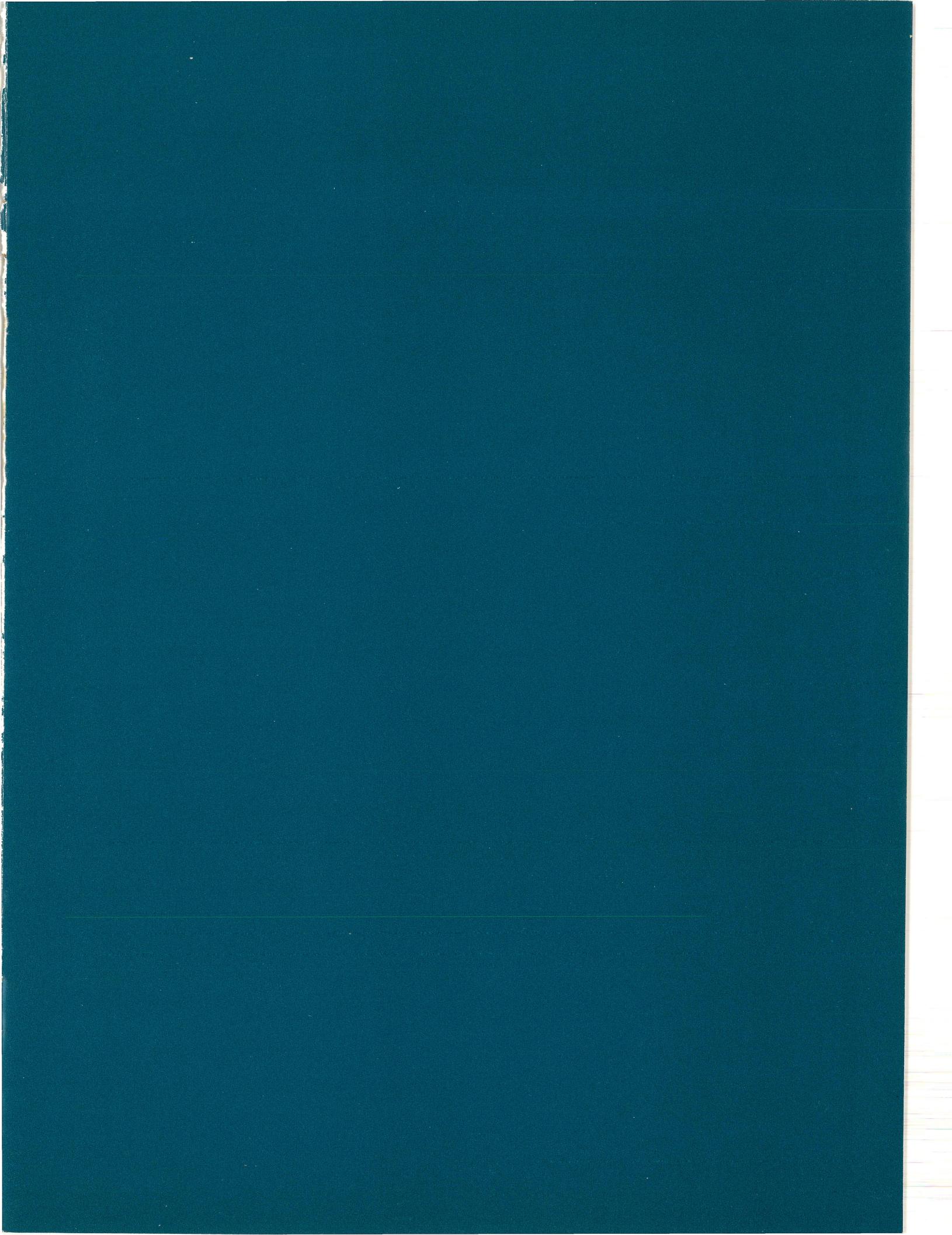
## Debugger and Profiler Features

The source-level Debugger allows the user to:

- Trace the entire program execution.
  - Execute code statement by statement.
  - Set breakpoints interactively and interrupt program execution at will.
  - Automatically execute stored commands at break-points.
  - Retain control over the program when run-time errors occur.
  - Print the history of statements leading to errors.
  - Watch variables or memory locations for unintended change.
  - Display the value of, and modify, variables using Pascal names.
  - Selectively print source Pascal statements.
- The Profiler produces a listing showing the number of times each statement in program is executed, isolating execution bottlenecks. With this data, the programmer can refine portions of code to substantially improve performance.



Our VAX/VMS Development Team celebrates its success.



Bob 281-9676

LORETTA:  
5017

Wayne

# Oregon Software

6915 S.W. Macadam Avenue  
Portland, Oregon 97219  
(503) 245-2202  
1-800-874-8501

Digital, PDP, VAX, VMS, RSTS, RT-II, RSX, Professional 350 and P/OS are trademarks of Digital Equipment Corporation. UNIX is a trademark of AT&T Bell Laboratories. TSX-Plus is a trademark of S&H Computing, Inc. VERSAdos is a trademark of Motorola, Inc. Pascal-2 is a trademark of Oregon Software.

#### International Distributors

**Australia:** Network Computer Software Pty. Ltd., Telephone: 211 2322; **Canada:** Prior Data Sciences Ltd., Telephone: 613-820-7235; **Denmark/Norway:** IMDATA A/S, Telephone: 02 63 22 33; **England:** JPY Associates, Ltd., Telephone: 01-949-1088/9; Real Time Products, Telephone: 01-588-0667; Software Sciences Ltd., Telephone: 252-544321; Special Software Ltd., Telephone: 889 271027; Unit C, Telephone: 903-205233; **Finland:** Mimarobe Oy, Telephone: 358-31-36333; **France:** YREL Electronique et Informatique, Telephone: 3-956 81 42; **Israel:** Triple-D Ltd., Telephone: 03-719195-6; **Japan:** ASR International Corporation, Telephone: 03-502-5550; **Rikei Corporation, Telephone:** 334-3949; **Mexico:** Internacional de Datos, Telephone: 905-554-4851; **Netherlands/Belgium:** Software Sciences, Telephone: 20-461331; **New Zealand:** Business Computers, Ltd., Telephone: 03-791-427; **Sweden/Norway:** Sern Dator Konsult AB, Telephone: 31-8308-00; **Switzerland:** ADCOMP AG, Telephone: 741-41-11; Muhlethaler Electronics AG Falken 100a, Telephone: 65-76-26-53; **W. Germany:** AC Copy, Telephone: 241-506096; Peripherie Computer Systems GmbH, Telephone: 089-678040.