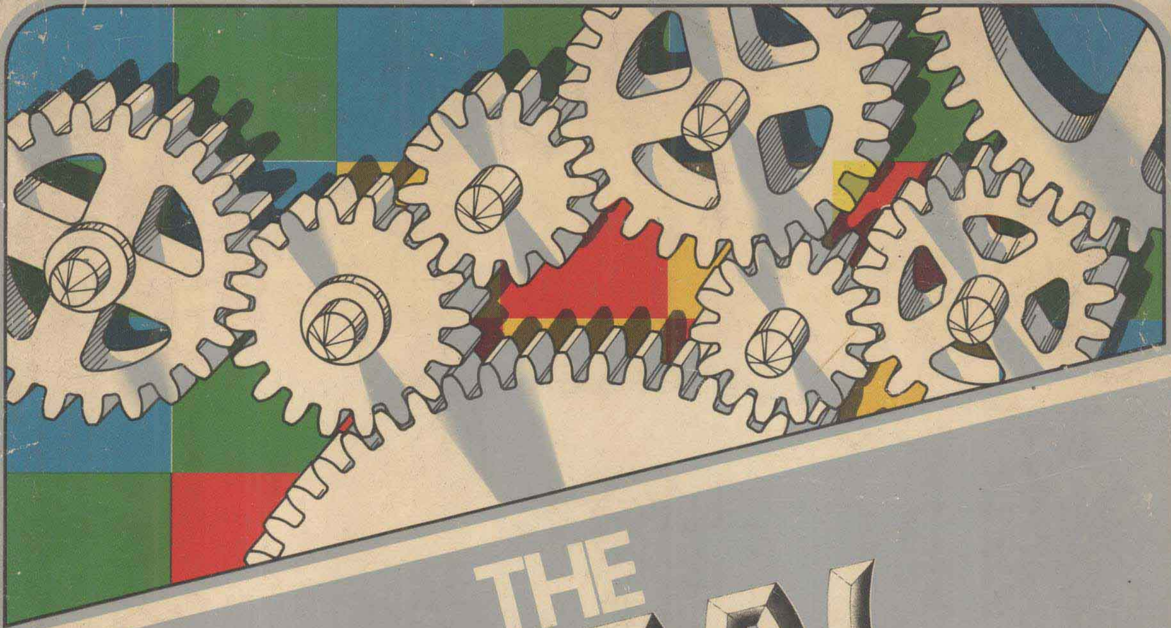
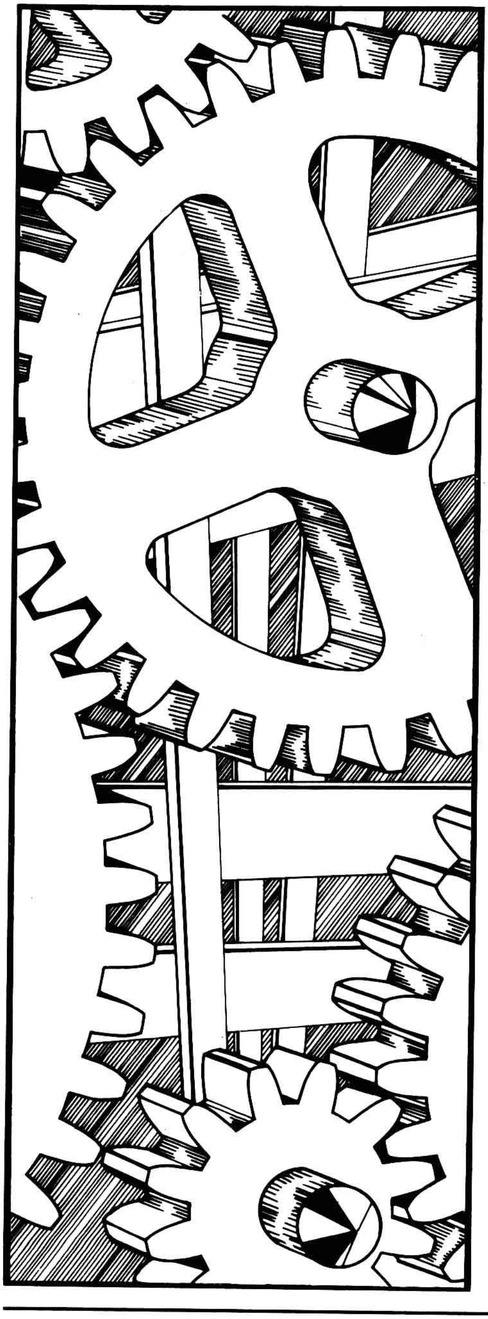


THE PASCAL HANDBOOK

JACQUES TIBERGHIEN

SYBEX



THE PASCAL HANDBOOK

JACQUES TIBERGHIEN

SYBEX

Every effort has been made to supply complete and accurate information. However, Sybex assumes no responsibility for its use, nor for any infringements of patents or other rights of third parties which would result.

Library of Congress Card Number: 80-53283
ISBN 0-89588-053-9
Printed in the United States of America
10 9 8 7 6 5 4 3 2

THE PASCAL HANDBOOK

Acknowledgements

Valuable contributions by many persons and organizations have made this book possible. The author is pleased to be able to acknowledge certain of them here.

Control Data, Hewlett Packard, Intel, Apple Computer, Microsoft, IBM, Oregon Minicomputer Software, University of California at San Diego and The Pascal User's Group provided important and timely information about the various Pascal implementations.

Eric Novikoff and Joseph Faletti were responsible for a detailed review of the work and many valuable improvements to the final manuscript.

Joseph Kalash and Charles Koester furnished valuable assistance in the development and verification of the program examples.

A. Winsor Brown of POINT 4 Data Corporation contributed many valuable revisions to the second printing of this book.

Julie Sickert and Janet Rampa supervised the manuscript and the editorial process, constantly contributing to the book's improvement.

Roger Gottlieb, Guy Orcutt, Karl Sterne and Rodnay Zaks provided guidance and support throughout the development and production of this book.

The entire Sybex staff accomplished the design, editing and timely production of THE PASCAL HANDBOOK in a most efficient and helpful manner.

Preface

When originally defined by Niklaus Wirth in 1971, Pascal was seen as a coherent, powerful and well-defined language and quickly gained wide acceptance. However, the very success of, and enthusiasm for, Pascal has led to the development of multiple and incompatible versions. These multiple versions have, in turn, created a worldwide effort at defining a new standard, the ISO (International Standards Organization) Standard. Despite this effort at standardization, new extensions are still being implemented.

As a result, most Pascal versions available today display some incompatibilities, and programs written in one version may not execute in another. When writing a Pascal program, a programmer must frequently verify and cross-check the definitions and effects of specific features being used.

Drawing together information scattered until now among diverse sources, this handbook has been written as a single reference manual designed to facilitate the use of Pascal by all Pascal users. Every feature of Pascal is explained in a convenient format for some major Pascal dialects including Jensen and Wirth's original definition, with the CDC implementation, the proposed ISO Standard, UCSD Pascal, Pascal 1000 (HP 1000), OMSI Pascal-1, and Pascal/Z. Compiler options are not included.

Organized alphabetically, this book is designed for ease in retrieving and understanding all features of the language. As such, it is a comprehensive and indispensible tool for the Pascal user.

Each significant feature of Pascal is a separate entry. Separate entries are found for:

— *symbols*, including:

reserved words	such as **PROGRAM, BEGIN** and **END**
operators	such as + – and *
delimiters	such as ; . ' and ( ).

— *predefined identifiers*, including:

types	such as REAL, INTEGER, CHAR
functions	such as ABS, SIN, SQR
procedures	such as READ, WRITE, GET
constants	such as MAXINT, TRUE, FALSE

— *concepts*, such as "global", "assignment" and "statement".

In order to provide easy access to information, a format with a quick-reference heading and up to four main sections has been consistently applied to the description of each entry.

The *heading* of each entry shows:

1. the feature being described in large, clear type with a brief definition
2. a chart indicating
   — whether the feature is a symbol, identifier, or concept
   — which versions of Pascal implement this feature.

The *description* of each entry is presented in four sections:

1. SYNTAX: a diagram showing how the entry may be legally used
2. DESCRIPTION: what it is, what it does
3. IMPLEMENTATION-DEPENDENT FEATURES: any differences between the various implementations, including appropriate syntax diagrams
4. EXAMPLES: whenever necessary, typical examples that clarify the use or meaning of the entry are presented.

The "Standard" referred to in this handbook is the *proposed ISO Standard* as published at this time (see reference list at the end of this book), not the original Jensen and Wirth definition of Pascal. This standard has not, however, been finalized and some features may still be modified. The author has therefore used his own judgment in the few cases where features were still under discussion. For example, *conformant arrays* are not described here, as they will probably not be included in the final standard.

Another practical problem that the author faced was to select the implementations to be described in this book. Some recent implementations are not widely used and some depart significantly from Standard Pascal. The author has selected widely used implementations that he feels conform to the original spirit of Pascal.

This book should provide a comprehensive and practical reference for Pascal programmers, whether novices or experienced users. By using a consistent and carefully organized format for the description of all Pascal features, this book will:

— encourage the use of all features of the language
— ease the programming process
— facilitate learning the language
— assist in the translation of programs from one version of Pascal into another.

The author hopes to have captured the current spirit of Pascal within the pages of this book and welcomes comments and suggestions for improvement.

HOW TO READ THIS BOOK

entry

A brief definition of the Pascal entry or feature is given here for quick reference.

- □ **SYMBOL**
- □ **IDENTIFIER**
- □ **CONCEPT**
- □ **STANDARD**
- □ **HP 1000**
- □ **J & W/CDC**
- □ **OMSI**
- □ **PASCAL/Z**
- □ **UCSD**

1
SYNTAX

The formal syntax of the entry is presented in diagram form or described in words. A syntax diagram illustrates the correct use of the word or symbol and reflects the rules for combining it with other legal constructs in a program. (See the next section for How to Read a Syntax Diagram.)

2
DESCRIPTION

What the entry means, what it is used for, and its effect in a program.

3
IMPLEMENTATION-DEPENDENT FEATURES

Enhancements or restrictions to the use of the entry that are specific to the various implementations of Pascal.

4
EXAMPLE

Whenever required, an example program featuring the Pascal entry is presented in order to:

- illustrate the correct syntax of the entry within an actual program
- show how it works with other elements of a program to accomplish a specific task.

ABOUT THE HEADING:

THE ENTRY BOX

The type-face used for the term in the entry box has a specific meaning:

BOLDFACE UPPERCASE	means that the term is a reserved word.
UPPERCASE	means that it is a predefined identifier or directive.
lowercase	means that it is a concept.

Note: Codes for ASCII characters are given in decimal.

THE CHART

1. The first column of the chart below the definition of the entry indicates the *type* of entry.

   SYMBOL refers to reserved words, symbols (operators and delimiters) or directives (context-dependent words) that have a fixed, predefined meaning in Pascal and cannot be redefined by the user.

   IDENTIFIER refers to words that have a predefined meaning in Pascal, but may be redefined by the programmer to take a different meaning. These include predefined program parameters, constants, types, procedures and functions.

   CONCEPT refers to general terms that describe or define the syntax and operation of the Pascal language. Unlike symbols and identifiers, concepts do not appear in programs.

entry

□ **STANDARD**
□ **HP 1000**
□ **J & W/CDC**
□ **OMSI**
□ **PASCAL/Z**
□ **UCSD**

2. The last three columns of the chart indicate in which *versions* of Pascal the entry has been implemented.

STANDARD is the proposed ISO Standard (DP 7185, first version).

HP 1000 refers to Pascal 1000 which was developed by Hewlett Packard for the HP 1000.

J&W/CDC refers to Jensen and Wirth's original definition of Pascal implemented on CDC 6000 series machines.

OMSI refers to OMSI Pascal-1, developed by Oregon Minicomputer Software, Inc. for the PDP-11.

PASCAL/Z is a Pascal compiler for Z-80 microcomputers distributed by Ithaca Intersystems, Inc.

UCSD refers to the II.0 Pascal system and language developed at the University of California, San Diego, for mini-micro-computer applications. Apple Pascal (1.0) is highly compatible with UCSD Pascal II.0.

HOW TO READ A SYNTAX DIAGRAM

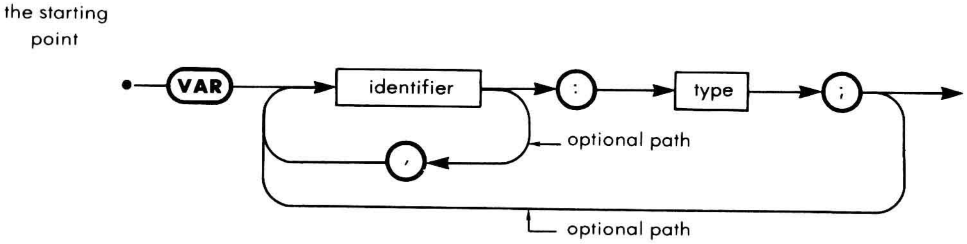
A box with rounded edges is used to represent predefined words including both reserved words **(BOLDFACE UPPERCASE)** and predefined identifiers (UPPERCASE). No blanks are permitted within such words.

A circle is used to represent reserved symbols (non-alphanumeric characters). No intervening blanks are permitted between the non-alphanumeric characters of a symbol.

A rectangle encloses syntax elements that are defined elsewhere in their own diagram. A qualifier, e.g., for type, may also be included.

Lines and arrows indicate authorized paths and are used to show the correct sequence of elements in the diagram. Except where noted in the text, blanks, comments and ends-of-lines may appear along the lines connecting enclosed words, symbols, and referenced syntax diagrams.

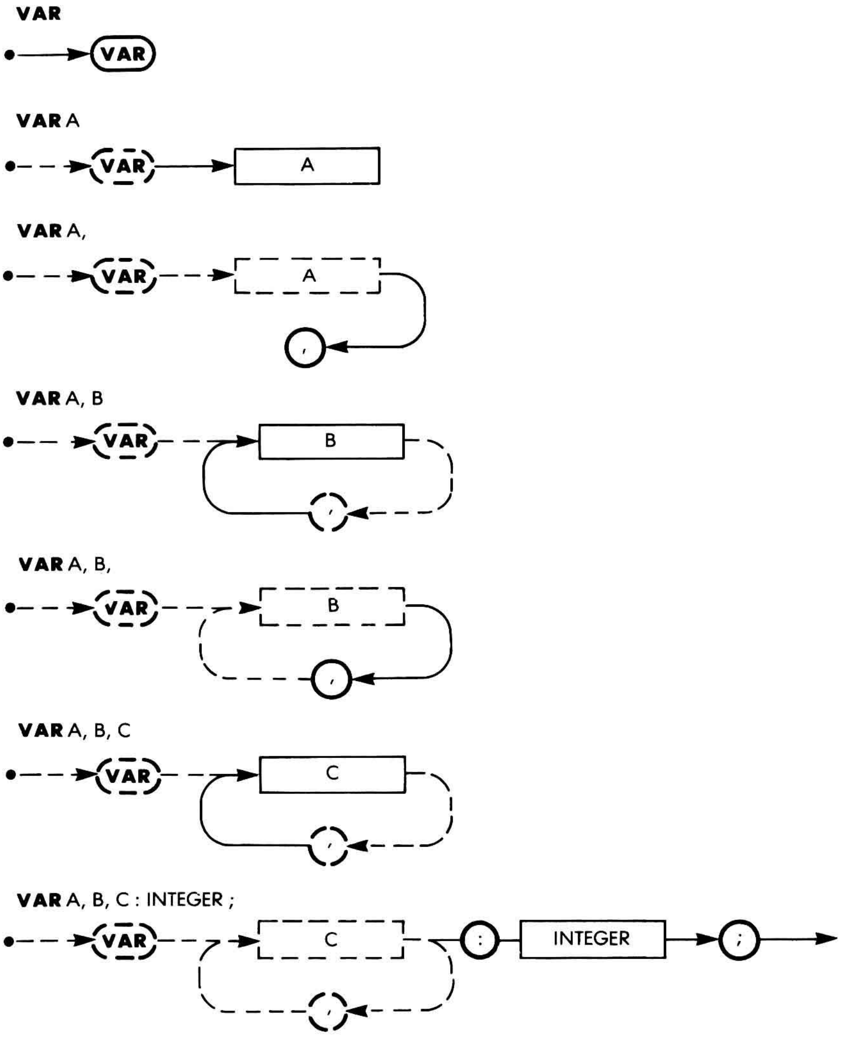
For example, the syntax for a VARiable declaration is drawn as follows:



The correctness of the declaration

VAR A,B,C : INTEGER;

can be verified by tracing through the syntax diagram. The diagram that follows is a step-by-step illustration of the way in which the VAR declaration is constructed (or verified) by following the syntax rules specified in the syntax diagram.



VAR

VAR A

VAR A,

VAR A, B

VAR A, B,

VAR A, B, C

VAR A, B, C : INTEGER ;

The SYBEX Library

BASIC PROGRAMS FOR SCIENTISTS AND ENGINEERS
by Alan R. Miller 340 pp., 120 illustr., Ref. B240
This second book in the "Programs for Scientists and Engineers" series provides a library of problem solving programs while developing proficiency in BASIC.

INSIDE BASIC GAMES
by Richard Mateosian 350 pp., 240 illustr., Ref. B245
Teaches interactive BASIC programming through games. Games are written in Microsoft BASIC and can run on the TRS-80, APPLE II and PET/CBM.

FIFTY BASIC EXERCISES
by J.P. Lamoitier 240 pp., 195 illustr., Ref. B250
Teaches BASIC by actual practice using graduated exercises drawn from everyday applications. All programs written in Microsoft BASIC.

YOUR FIRST COMPUTER
by Rodnay Zaks 260 pp., 150 illustr., Ref. C200A
The most popular introduction to small computers and their peripherals: what they do and how to buy one.

DON'T (or How to Care for Your Computer)
by Rodnay Zaks 220 pp., 100 illustr., Ref. C400
The correct way to handle and care for all elements of a computer system including what to do when something doesn't work.

INTRODUCTION TO WORD PROCESSING
by Hal Glatzer 200 pp., 70 illlustr., Ref. W101
Explains in plain language what a word processor can do, how it improves productivity, how to use a word processor and how to buy one wisely.

INTRODUCTION TO WORDSTAR
by Arthur Naiman 200 pp., 30 illustr., Ref. W110
Makes it easy to learn how to use WordStar, a powerful word processing program for personal computers.

FROM CHIPS TO SYSTEMS: AN INTRODUCTION TO MICROPROCESSORS
by Rodnay Zaks 560 pp., 255 illustr., Ref. C201A
A simple and comprehensive introduction to microprocessors from both a hardware and software standpoint: what they are, how they operate, how to assemble them into a complete system.

MICROPROCESSOR INTERFACING TECHNIQUES
by Rodnay Zaks and Austin Lesea 460 pp., 400 illustr., Ref. C207
Complete hardware and software interconnect techniques including D to A conversion, peripherals, standard buses and troubleshooting.

PROGRAMMING THE 6502
by Rodnay Zaks 390 pp., 160 illustr., Ref. C202
Assembly language programming for the 6502, from basic concepts to advanced data structures.

6502 APPLICATIONS BOOK

by Rodnay Zaks 280 pp., 205 illustr., Ref. D302
Real life application techniques: the input/output book for the 6502.

6502 GAMES

by Rodnay Zaks 300 pp., 140 illustr., Ref. G402
Third in the 6502 series. Teaches more advanced programming techniques, using games as a framework for learning.

PROGRAMMING THE Z80

by Rodnay Zaks 620 pp., 200 illustr., Ref. C280
A complete course in programming the Z80 microprocessor and a thorough introduction to assembly language.

PROGRAMMING THE Z8000

by Richard Mateosian 300 pp., 125 illustr., Ref. C281
How to program the Z8000 16-bit microprocessor. Includes a description of the architecture and function of the Z8000 and its family of support chips.

THE CP/M HANDBOOK (with MP/M)

by Rodnay Zaks 330 pp., 100 illustr., Ref. C300
An indispensable reference and guide to CP/M — the most widely used operating system for small computers.

INTRODUCTION TO PASCAL (Including UCSD PASCAL)

by Rodnay Zaks 420 pp., 130 illustr., Ref. P310
A step-by-step introduction for anyone wanting to learn the Pascal language. Describes UCSD and Standard Pascals. No technical background is assumed.

THE PASCAL HANDBOOK

by Jacques Tiberghien 490 pp., 350 illustr., Ref. P320
A dictionary of the Pascal language, defining every reserved word, operator, procedure and function found in all major versions of Pascal.

PASCAL PROGRAMS FOR SCIENTISTS AND ENGINEERS

by Alan Miller 400 pp., 80 illustr., Ref. P340
A comprehensive collection of frequently used algorithms for scientific and technical applications, programmed in Pascal. Includes such programs as curve-fitting, integrals and statistical techniques.

50 PASCAL PROGRAMS

by Rudolph Langer and Rodnay Zaks 275 pp., 90 illustr., Ref. P350
A collection of 50 Pascal programs ranging from mathematics to business and games programs. Explains programming techniques and provides actual practice.

APPLE PASCAL GAMES

by Douglas Hergert and Joseph T. Kalash 380 pp., 40 illustr., Ref. P360
A collection of the most popular computer games in Pascal challenging the reader not only to play but to investigate how games are implemented on the computer.

INTRODUCTION TO UCSD PASCAL SYSTEMS

by Charles T. Grant and Jon Butah 300 pp., 110 illustr., Ref. P370
A simple, clear introduction to the UCSD Pascal Operating System for beginners through experienced programmers.

INTERNATIONAL MICROCOMPUTER DICTIONARY

140 pp., Ref. X2

All the definitions and acronyms of microcomputer jargon defined in a handy pocket-size edition. Includes translations of the most popular terms into ten languages.

MICROPROGRAMMED APL IMPLEMENTATION

by Rodnay Zaks 350 pp., Ref. Z10

An expert-level text presenting the complete conceptual analysis and design of an APL interpreter, and actual listings of the microcode.

SELF STUDY COURSES

Recorded live at seminars given by recognized professionals in the microprocessor field.

INTRODUCTORY SHORT COURSES:

Each includes two cassettes plus special coordinated workbook (2½ hours).

S10—INTRODUCTION TO PERSONAL AND BUSINESS COMPUTING

A comprehensive introduction to small computer systems for those planning to use or buy one, including peripherals and pitfalls.

S1—INTRODUCTION TO MICROPROCESSORS

How microprocessors work, including basic concepts, applications, advantages and disadvantages.

S2—PROGRAMMING MICROPROCESSORS

The companion to S1. How to program any standard microprocessor, and how it operates internally. Requires a basic understanding of microprocessors.

S3—DESIGNING A MICROPROCESSOR SYSTEM

Learn how to interconnect a complete system, wire by wire. Techniques discussed are applicable to all standard microprocessors.

INTRODUCTORY COMPREHENSIVE COURSES:

Each includes a 300-500 page seminar book and seven or eight C90 cassettes.

SB1—MICROPROCESSORS

This seminar teaches all aspects of microprocessors: from the operation of an MPU to the complete interconnect of a system. The basic hardware course (12 hours).

SB2—MICROPROCESSOR PROGRAMMING

The basic software course: step by step through all the important aspects of microcomputer programming (10 hours).

ADVANCED COURSES:

Each includes a 300-500 page workbook and three or four C90 cassettes.

SB3—SEVERE ENVIRONMENT/MILITARY MICROPROCESSOR SYSTEMS

Complete discussion of constraints, techniques and systems for severe environment applications, including Hughes, Raytheon, Actron and other militarized systems (6 hours).

SB5—BIT-SLICE

Learn how to build a complete system with bit slices. Also examines innovative applications of bit slice techniques (6 hours).

SB6—INDUSTRIAL MICROPROCESSOR SYSTEMS

Seminar examines actual industrial hardware and software techniques, components, programs and cost (4½ hours).

SB7—MICROPROCESSOR INTERFACING

Explains how to assemble, interface and interconnect a system (6 hours).

SOFTWARE

BAS 65™ CROSS-ASSEMBLER IN BASIC

8″ diskette, Ref. BAS 65
A complete assembler for the 6502, written in standard Microsoft BASIC under CP/M® .

8080 SIMULATORS

Turns any 6502 into an 8080. Two versions are available for APPLE II.

APPLE II cassette, Ref. S6580-APL(T)
APPLE II diskette, Ref. S6580-APL(D)

FOR A COMPLETE CATALOG OF OUR PUBLICATIONS

U.S.A.
2344 Sixth Street
Berkeley,
California 94710
Tel: (415) 848-8233
Telex: 336311

SYBEX-EUROPE
4 Place Felix Eboue
75583 Paris Cedex 12
Tel: 1/347-30-20
Telex: 211801

SYBEX-VERLAG
Heyestr. 22
4000 Düsseldorf 12
West Germany
Tel: (0211) 287066
Telex: 08 588 163

SYBEX