Personal Computer
Computer Language
Series

Pascal Compiler

by Microsoft

Personal Computer
Computer Language
Series

Pascal Compiler

by Microsoft

First Edition (August 1981)

Changes are periodically made to the information herein; these changes will be incorporated in new editions of this publication.

Products are not stocked at the address below. Requests for copies of this product and for technical information about the system should be made to your authorized IBM Personal Computer Dealer.

A Product Comment Form is provided at the back of this publication. If this form has been removed, address comment to: IBM Corp., Personal Computer, P.O. Box 1328, Boca Raton, Florida 33432. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligations whatever.

International Business Machines Corporation Boca Raton, Florida 33432

IBM Program License Agreement

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE OPENING THIS DISKETTE(S) OR CASSETTE(S) PACKAGE. OPENING THIS DISKETTE(S) OR CASSETTE(S) PACKAGE INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD PROMPTLY RETURN THE PACKAGE UNOPENED; AND YOUR MONEY WILL BE REFUNDED.

IBM provides this program and licenses its use in the United States and Puerto Rico. You assume responsibility for the selection of the program to achieve your intended results, and for the installation, use and results obtained from the program.

LICENSE

You may:

a. use the program on a single machine;

b. copy the program into any machine readable or printed form for backup or modification purposes in support of your use of the program on the single machine (Certain programs, however, may include mechanisms to limit or inhibit copying. They are marked "copy protected.");

c. modify the program and/or merge it into another program for your use on the single machine (Any portion of this program merged into another program will continue to be subject to the terms and conditions of this Agreement.); and,

d. transfer the program and license to another party if the other party agrees to accept the terms and conditions of this Agreement. If you transfer the program, you must at the same time either transfer all copies whether in printed or machine-readable form to the same party or destroy any copies not transferred; this includes all modifications and portions of the program contained or merged into other programs.

You must reproduce and include the copyright notice on any copy, modification or portion merged into another program.

YOU MAY NOT USE, COPY, MODIFY, OR TRANSFER THE PROGRAM, OR ANY COPY, MODIFICATION OR MERGED PORTION, IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS LICENSE.

IF YOU TRANSFER POSSESSION OF ANY COPY, MODIFICATION OR MERGED PORTION OF THE PROGRAM TO ANOTHER PARTY, YOUR LICENSE IS AUTOMATICALLY TERMINATED.

TERM

The license is effective until terminated. You may terminate it at any other time by destroying the program together with all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this Agreement or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy the program together with all copies, modifications and merged portions in any form.

LIMITED WARRANTY

THE PROGRAM IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU (AND NOT IBM OR AN AUTHORIZED PERSONAL COMPUTER DEALER) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Continued on inside back cover

CONTENTS

CHAPTER 1. INTRODUCTION

Contents

IBM Personal Computer Pascal

This document describes IBM Personal Computer Pascal. We have assumed that you have a general knowledge of Pascal from such publications as "Pascal User Manual and Report" by Jensen and Wirth, "Introduction to Pascal" by Welsh and Elder, or from other sources.

The Pascal Language

Pascal was originally designed by Niklaus Wirth with two primary goals: to teach programming as a systematic discipline, and to implement programs in a reliable and efficient way. But there are other reasons for the widespread interest in Pascal as a general purpose programming language and as a system program implementation language.

- Pascal, a relatively modern language, has benefited from many earlier languages (such as ALGOL) and is forming the basis of several new ones.

- The key emphasis of Pascal is its role as a higher level language; that is, a language providing useful abstract tools for specifying data structures and algorithms, independent of the underlying implementation.

- The user need not be concerned with the representation of data (the number of bytes per variable, organization of arrays, address size, and so on).

- The programmer can more accurately specify the characteristics of variables, such as the range of values allowed (VAR I: 0..99) or decide whether

to trade space for time in accessing variable components. (PACKED keyword)

IBM has added several goals to this list:

1. IBM Personal Computer Pascal is designed to be a systems implementation language, especially suitable for writing compilers, interpreters, operating systems, and so on.

2. Generating efficient code is paramount. The various language extensions, and the global optimizer phase (pass two of the compiler) all work toward minimizing the time and space needed by compiled programs.

3. Operations that are done easily in assembly language should be easy to do in IBM Personal Computer Pascal.

IBM Personal Computer Pascal generally conforms to the ISO draft (ISO/TC 97/SC 5 N595).

IBM's intent is that correct standard programs compile and run correctly on the IBM Personal Computer Pascal compiler without changes. Since IBM features introduce new reserved words and other elements, both the IBM features and extensions are summarized in Appendix E.

IBM Personal Computer Pascal Extensions

The extensions include:

- Compiler directives
- Attributes
- Super array type
- String processing with CONCAT
- Constant values
- Systems implementation extensions

Compiler Directives

We provide control over generating error checking code, listing format controls, a construct for conditional compilation, and a mechanism for inserting other source files into a compilation. There are 30 of these "metalanguage" commands available.

Unit

Pascal is an excellent language for very large programs that must be reliable (such as system software applications), and IBM Personal Computer Pascal supports separtely compiled modules using the concept of a unit.

A *unit* is a set of related data types, variables, constants, procedures, and functions, plus an initialization procedure. It has two parts: the interface and the implementation.

The *interface* contains a list of identifiers to export and their declarations (including procedure and function headers).

The *implementation* contains any local declarations, procedure and function bodies, and the initialization code. Some unit routines may be implemented in assembly code instead of Pascal.

A program (or implementation or other interface) can use a unit, giving the interface but not the implementation. This method provides a more structured way of breaking a program into modules than external procedures and variables.

Attributes

Attributes for variables, procedures, and functions give you control at the linker text level.

The attributes include:

- PUBLIC and EXTERN for global identifier linking
- STATIC and READONLY for variables and PURE for procedures and functions.

Super Array

IBM Personal Computer Pascal provides a *super array* type, to let array lengths vary. With the super array type, the lower bound is given but the upper bound is undefined.

Although the super array type cannot be used directly for variables (that is, VAR V = SUPER ARRAY [0..*] of REAL:), it can be used in two important ways:

- As the type of a formal reference parameter
- As the referent type of a pointer.

In addition, any variable can be given a type derived from the super array type using a "designator". For example:

```
TYPE
  VECT = SUPER ARRAY [0..*] OF REAL;
VAR
  PVEC: ^VECT; V10: VECT (10);
PROCEDURE SORT (VAR V: VECT);
BEGIN
  NEW (PVEC, UPPER (V));
    .
    .
    .
END;
    .
    .
    .
SORT (V10);
```

Here VECT is the super array type. PVEC is a pointer to an instance of the type.

The NEW allocates a new VECT with the upper bound given in the second parameter, in this case the same upper bound as the parameter V.

V10 is a variable, an instance of a VECT with an upper bound of 10. The parameter V can take a variable of any type derived from VECT; for example, V10 or PVEC^.

The super array concept handles both dynamic and conformant arrays in a clean and efficient way.