READ ME FIRST!

YOUR REGISTRATION

Every product that Crescent Software sells includes a registration card with serial number. You will need the serial number to receive technical support from us, or to purchase upgrades at a reduced price. Therefore, we recommend that you write this number on your disk labels *and* your manual. Upgrades do not include a new serial number, so the original serial number should be copied onto your upgrade disks.

Please note that when a product is purchased directly from us, we remove the mail-in portion of the registration card. This saves both you and us from having to deal with yet more paperwork. However, please notify us if the card has been removed and the package was not addressed directly to you. This way you can be registered as the customer entitled to technical support.

LICENSING

All Crescent Software products are copyrighted. We include the source code for your enlightenment and convenience. You may *not* distribute any of our source code or .OBJ or .LIB or any other files in such a manner that someone else could use them. If you choose to distribute your own source code, we applaud you even if it clearly shows how to call our routines. But if your customer requires our source code or the ability to recompile your program, they will also need their own copy of our product. Alternatively, we can license a subset of the routines. Pricing depends on which routines and how many. Please call our office for more details.

BEFORE YOU CALL FOR SUPPORT

Crescent Software provides knowledgeable technical assistance for all of our products, but *only* to legitimate owners. Further, we will support only the person to whom the product is registered—you will be asked to furnish your product serial number and name when you call. We will gladly answer reasonable questions about our products, as well as questions about BASIC programming in general. However, you should read our manuals as well as those that came with your BASIC compiler before calling us. We also include many demonstration programs that show how to use the various routines in our products, and you should refer to those as well, before calling. Technical support is available Monday through Friday, from 9:00 A.M. through 5:30 P.M. Eastern time.

Due to the large volume of support calls we receive each day, we ask that you try to be specific and get right to the details of your problem. If you are receiving error messages from QuickBASIC or LINK, try to be at your PC with the program loaded, if at all possible. Otherwise, please be prepared to relate the *exact* wording of the message. Also, be sure to have your product version number ready. The version number for all Crescent product is kept in the disk's volume label:

```
You Type:        VOL A:
Dos responds:    Volume in drive A is QPPro 4.xx
```

In addition to the error messages and possible causes listed in the QuickPak Professional manual, the following issues occur frequently with that product.

1. When developing very large programs in the QuickBASIC or QBX environment, you may receive "Out of string space" or "Out of memory" messages. At some point, of course, your program will become too large to fit entirely in memory. However, the MAKEQLB utility we provide with QuickPak Professional can also help. MAKEQLB is listed in the owner's manual, and it will create a subset of the PRO.QLB Quick Library containing only the routines your program actually needs.

2. Because of the enormous number of programs that are included with QuickPak Professional, they are distributed in compressed (.ZIP) format. The PKUNZIP utility program for decompressing these files is included on disk 1 of the distribution disks. The installation program (INSTALL.EXE) automatically invokes PKUNZIP to decompress the selected .ZIP files, but you can also use PKUNZIP manually to decompress the supplied .ZIP files as follows:

```
PKUNZIP FILENAME.ZIP [FILENAME.EXT]
```

You may extract individual files from the supplied .ZIP files by entering the name of the desired file after the name of the .ZIP file.

3. We provide the assembly language source code for all of our routines, but it is very unlikely that you will need to reassemble the source code to create .OBJ files. If you simply need an object module for linking or other purposes, please see the sections entitled "Extracting object modules from PRO.LIB" in the owner's manual. If you really do need to modify and assemble a routine, please read the MASM51.DOC file that is extracted with the assembly source code.

4. If you are working the QuickBASIC environment and receive a "File not found" error and the cursor is placed on an $INCLUDE statement, you probably loaded one of our demonstration programs from a directory that is not the current default. Although QuickBASIC will look for related modules in the directory from with the main program was loaded, it will not do this with Include files. We recommend that you start QuickBASIC, and then change to the directory you plan to work in before loading the program. Alternatively, the following sample of QB.BAT batch file may be placed in your working directory:

```
\QP45\QB %1 /L \CRESCENT\PRO /AH
```

This file assumes that QB.EXE is in a directory named \QB45, and that our PRO.QLB file is in a directory named \CRESCENT. Then to begin a work session you would start QuickBASIC from your working directory like this:

```
QB program
```

5. If you are compiling a multi-module program from the DOS command line, it is essential that you compile each BASIC source file, and then link them all together. If your are not sure which files are necessary (possibly when compiling one of our demos), look at the main program's Make file. This file has the same name as the main BASIC program, but with a .MAK extension.

6. If you receive a "String space corrupt" error, use the "Add watch" capability to watch the expression FRE(""), and then step through your program using the F10 key. This will cause QuickBASIC to evaluate the integrity of its string memory at each statement.