

A1. Installation and Use of VPLX on an IBM 386/486/Pentium PC

A1.1 Introduction

The file VPLX.EXE represents an executable program for 386 or larger IBM PC's with a minimum of 8 megabytes of memory, under DOS or Windows 3.1. VPLX.EXE was compiled by the Microsoft FORTRAN PowerStation compiler and uses 32-bit addressing, thus depending on extended memory. The program requires a math coprocessor (a separate component for 386 and 486SX machines, but built into 486DX, DX2 and PENTIUM chips). (Based on Microsoft documentation for the FORTRAN PowerStation, VPLX.EXE will probably not execute without a coprocessor or its equivalent. The author may be able to produce a version to run without a coprocessor by resetting the compiler options. However, VPLX is likely to be slower by a factor of 10 without a coprocessor.)

Any executable files carry the potential for distributing viruses. The author checks disks for viruses before mailing, but users should consider checking any copy of VPLX.EXE and other files for viruses. The author is unaware of any feature of VPLX that could be mistaken for a virus. Consequently, any indication by virus protection software that a virus is present should be taken entirely at face value.

Two critical files are distributed with VPLX, DOSXNT.386 and DOSXMSF.EXE. Both are proprietary products of Microsoft and its partners. These files must not be altered, reverse engineered, or built directly into other software, but they may be copied and distributed for use with VPLX and any related programs developed for use with VPLX (e.g., FILEFIXV, described below).

A1.2 Installation

General remark: The following directions assume some familiarity with DOS, and some of the following assumes DOS 5.0 or higher, although other approaches may be taken to install the program with earlier versions.

1) First check the setting of FILES in your C:\CONFIG.SYS file. (You may do so in DOS by EDIT C:\CONFIG.SYS, if you are familiar with the DOS editor EDIT. You may also type MORE < C:\CONFIG.SYS at the DOS prompt.) Running VPLX requires at least FILES=50, a general requirement of the Microsoft FORTRAN PowerStation. If your CONFIG.SYS satisfies this requirement, no changes are necessary.

A1.2

Otherwise, first copy CONFIG.SYS to another file, e.g., CONFIG.BAK (COPY CONFIG.SYS CONFIG.BAK) and edit CONFIG.SYS to make this change to FILES=50. (For example, use EDIT or the Program Window of PC-SAS run interactively. You may also use a word processor but be sure to write the file out as an ASCII file instead of in the default file format of the word processor. For example, both WordPerfect and Word, can also output an ASCII file if specifically asked to, but each uses its own file format as the default. Changing CONFIG.SYS into something other than an ASCII file, however, would have severe effects on the performance of the computer.)

2) Starting with Version 95.07, VPLX will have an associated ASCII file VPLERROR.TXT, which will provide detailed information on some of the error messages generated by VPLX. This file may be printed or examined to interpret what a specific VPLX error code means. The most advantageous use of the file, however, is to copy it into C:\VPLX\VPLERROR.TXT. If an error occurs with an associated error code, VPLX will look for this file and print the associated error message, if possible.

In DOS, create a directory C:\VPLX by the following commands:

```
C:  
CD \  
MKDIR VPLX
```

You may also use the File Manager of WINDOWS to create the directory. From DOS, use COPY to copy the file from the diskette into C:\VPLX. For example,

```
COPY A:\VPLERROR.TXT C:\VPLX
```

This operation could also be done with the File Manager.

- 3) Select a directory on the hard drive of the PC. Either of the two strategies are recommended:
- 1) Use C:\VPLX. This strategy helps to record where VPLX.EXE actually sits, but will require modification of the DOS path.
 - 2) Select a directory on the current DOS path. The path is almost surely set in the C:\AUTOEXEC.BAT file. You may type PATH at the DOS prompt to see the current path assignment, which lists a series of directory that DOS will search to find executables not in the current directory.

If the first strategy is selected, then you must edit the PATH statement in the C:\AUTOEXEC.BAT file to include the new directory, again using an editor to produce an ASCII output, just as for CONFIG.SYS. For example, the current statement may be:

```
PATH C:\WINDOWS;C:\DOS
```

which you should modify by adding C:\VPLX to the end of the list.

```
PATH C:\WINDOWS;C:\DOS;C:\VPLX
```

The actual statement may be longer than his example. Note the use of ";" as a delimiter between directories. If all else fails, consult DOS documentation.

If VPLX and DOSXMSF.EXE are not on the current path, then it may be impossible to start VPLX, VPLX may start but produce an error message that it cannot find DOSXMSF.EXE, or the PC may completely lock up. Thus, revising the DOS path is a critical part of installation.

A possible problem may occur if other software is installed in such a way that it modifies the path, particularly through a .BAT file. Thus, if the other software is run before using VPLX, the current path may not be the one specified by C:\AUTOEXEC.BAT. If VPLX locks up or produces any of these problems appear, check the path by typing PATH at the DOS prompt before attempting to run VPLX.

4) Copy VPLX.EXE, DOSXNT.386, and DOSXMSF.EXE to the selected directory on the path. For example,

```
COPY A:\VPLX.EXE C:\VPLX
COPY A:\DOSXNT.386 C:\VPLX
COPY A:\DOSXMSF.EXE C:\VPLX
```

5) A further step is required to run VPLX within a DOS window within Windows. The directory into which Windows has been installed, probably C:\WINDOWS, contains a file SYSTEM.INI. This file must be modified, but before doing so, make a copy, e.g., SYSTEM.BAK, just in case anything goes wrong as you edit the file. Again, you may use EDIT, SAS, or any other editor that writes a standard DOS file as an output. (Do not change SYSTEM.INI into, for example, a Word or WordPerfect file.) The SYSTEM.INI file includes a section beginning

```
[ 386Enh ]
```

Add to this section of SYSTEM.INI the line:

A1.4

```
device=c:\vplx\dosxnt.386
```

Of course, if you have copied DOSXNT.386 into some other directory, then modify the previous line accordingly.

6) If any changes have been made to C:\CONFIG.SYS, C:\AUTOEXEC.BAT, or SYSTEM.INI, reboot the machine, [CONTROL]-[ALT]-[DEL] simultaneously, to have the changes take effect.

A1.3 Running VPLX from DOS or Within a DOS Window

Once VPLX has been installed, then it may be started from any directory. The usual approach to running VPLX is to prepare an ASCII file of commands for input to VPLX and to have the output sent to another file. For example,

```
vplx < exam1.crd > exam1.lis
```

will read commands from exam1.crd and write them to exam1.lis. The output file includes special characters that work with the DOS PRINT command, so that it may be possible to

```
print exam1.lis
```

Otherwise, the output file may be read into a word processor and edited for display. It is possible to take a look at the file with the DOS command

```
more < exam1.lis
```

Again, an ASCII editor, such as EDIT, should be used for preparing the input file, for example, exam1.crd. As commented earlier, word processors usually produce non-ASCII files with their own control information, but most, including WordPerfect, will output ASCII files on special request..

Most of the arrays in VPLX.EXE except the double precision work array are dimensioned at their "mainframe" sizes - the same sizes that the author uses on the Census Bureau's larger VAX and SUN systems. The author works primary on PC's with 8 megabytes of memory or more, and at least 6 megabytes is required. The size of the double precision array may be in the neighborhood of 100,000 - 160,000 double precision words. (In the larger "mainframe" version, the array is usually of size 1,000,000.) A PC with 8 megabytes may accommodate approximately 500,000, and specially compiled versions may be available upon request for this larger size if justified by a large application.

The first time VPLX starts during a session from a DOS prompt, nothing may seem to be happening except for hard disk accesses for perhaps 5-10 seconds. This is especially noticeable if the output is sent directly to the screen through:

```
vplx < exam1.crd
```

The author suspects that the DOS extender is rearranging extended memory for its own use or performing some related housekeeping function. After this short interval, VPLX then prints the initial message and begins. Users should allow at least 10 seconds before concluding that a lock up or other malfunction has occurred.

If there are troubles with available memory, try closing other active applications before starting VPLX.

VPLX will also work from commands typed directly from the keyboard, which is advantageous for simple operations such as CONTENTS or a short DISPLAY. It is also possible to INCLUDE files. After typing the last command, an end-of-file, CONTROL-Z, (^Z), followed by ENTER, is required. There are three variants:

```
vplx > exam1.lis
```

will write the output to exam1.lis ;

```
vplx
```

will send results directly to the screen, with no control over scrolling; and

```
vplx | more
```

will enable scrolling through the output. With the last variant, the output begins to appear after the end-of-file, ^Z, is entered. ^Z is necessary if one has included a file and left VPLX looking for any possible further instructions on the last step. CONTROL-C, (^C), may be used to halt VPLX, canceling further operations, when VPLX is reading input from the keyboard.

A1.4 Special Considerations in Reading Data Files

Some word processors and other software, including WordPerfect and PC-SAS, write a final record with the single character ^Z (CONTROL Z, i.e. the ASCII character with decimal value 26, hexadecimal 1A). Previous Microsoft Fortrans returned end-of-file on reaching this ending character, but the PowerStation returns this character as data, filling remaining characters with blanks. Beginning with version 93.12, VPLX looks for this situation on input command files,

A1.6

such as exam1.crd above, so an ending ^Z should have no effect. Unfortunately, a problem will occur for reading data files, using a FORMAT specified by the user. If the format includes the first character as part of a data item, then a read error occurs. VPLX reports this error but continues to process the file, giving suitable results. If, on the other hand, the format skips the first character, then the situation is actually worse: no read error occurs on reaching ^Z, but VPLX will receive 0's for each of the input variables. The end-of-file is returned to VPLX on the next read. Under some replication options, for example, the jackknife, or when using one or more BY variables, VPLX will typically report that the file is out of order and stop. If VPLX does not abruptly stop for these reasons, then more harm will be done for unweighted analysis, because an extraneous observation has been added to the data set, than weighted analysis, because the weight, like all of the other variables, will be zero.

The author has notified Microsoft of the benefit of offering an alternative method to handle this situation, such as a non-standard parameter in the FORTRAN OPEN statement requesting that an end of file condition be returned when encountering ^Z. In the meantime, there are two remedies for this situation:

- 1) When reading input, always include the first character in the format specification, thus ensuring that VPLX will stop reading the file. It may be necessary to create a variable that can otherwise be ignored.
- 2) Run a utility FILEFIXV, which is included separately. This file should be copied to the same directory as VPLX.

FILEFIXV reads an ASCII file that may possibly end with ^Z and writes an output file, with a different name, without the ending ^Z. To run FILEFIXV, type FILEFIXV at the DOS prompt. The program will interactively request 3 pieces of information:

- 1) The name of the file to be changed;
- 2) The name of the new output file;
- 3) The maximum number of characters in any one record (not more than 32,000).

If in doubt, provide a somewhat generous upper bound for 3) - if the value given is less than the actual maximum length, then FILEFIXV will lose characters beyond the maximum length. On the other hand, routinely responding to the last question with the maximum of 32,000 will result in much slower execution than if a reasonable value is given, for example if 80 is appropriate. FILEFIXV will read in the specified number of characters (possibly filled with ending blanks if shorter than the stated maximum) and write the record back out, trimming off any ending blanks. Possibly, the resulting file will be shorter than the input because of this elimination of trailing blanks, but the file will still be successfully read by VPLX because of FORTRAN's convention of effectively filling incoming records with blanks when the actual record is shorter than the specified format.

Note that the output file from FILEFIXV may not be equally useful for all other purposes. For example, if FILEFIXV shortens records, the default option in the data step of SAS does not assume trailing blanks but reads from the next record instead. The SAS data step option MISSEVER is required to instruct the data step to use the FORTRAN convention.

FILEFIXV also uses DOSXMSF.EXE and, in addition, under Windows, DOSEXT.386. If these files are already installed for VPLX, then nothing further should be required to run FILEFIXV.