NONMEM and PDx-Pop Support Guide
Original Document: November 27, 2006
Document Revision: Original

William J. Bachman, Ph.D.
Director, Pharmacometrics R&D
Icon Development Solutions
6031 University Blvd., Suite 300
Ellicott City, MD 21076
Office 410-696-3002
Remote 301-330-2576
Cell 301-467-8635
bachmanw@iconus.com

Introduction:

The biggest support issue for both NONMEM and PDx-Pop is the improper or incomplete installation of the FORTRAN compiler that is required by both programs. The issue involves the system environment variables required by the compiler and, in the case of Digital/Compaq/Intel series of compilers, the Microsoft Software Development Kit (SDK) which is generally provided by Microsoft Visual Studio software. This document will address proper installation and trouble-shooting of compiler installations for the major compilers that IDS can support for NONMEM and PDx-Pop. In general, if NONMEM can be run from the command line (a standard DOS window, e.g. one accessed by "All Programs > Accessories > Command Prompt" in Windows XP), then the NONMEM installation will work with PDx-Pop.

Digital/Compaq/Intel Compiler Issues:

The earlier compilers (Digital and Compaq) were often successful at setting the environment variables for both the compiler and the SDK when request to do so by the user during installation of the compiler. The Intel compiler often does not set the system environment variables even when requested to do so by the user during setup. What actually happens is that the compiler installation sets the environment for a "DOS-like" environment called a Build Environment that can be accessed from the Program Menu for the compiler program that the users confuse for a "standard" DOS window which requires the system environment variables to be set for it. NONMEM can be installed and run from this Build Environment. However, NONMEM cannot be run from a standard DOS window if it has been installed from the Build Environment window (because the standard window needs the system environment variables to be set for it). Also, PDx-Pop cannot use this "Build Environment"-installed NONMEM because it also needs the correct system environment variables. Users often state that "PDx-Pop does not work" when in actuality it is the NONMEM installation using the Build Environment that is the real problem.

Keep in mind that there are two sets of environment variables that are required for these compilers to operate successfully with NONMEM at a standard DOS

window (and that if NONMEM runs at a standard DOS window, PDx-Pop will also properly run NONMEM). The compiler and the SDK installations provide batch files to guide you in setting the system environment variables. If both of these batch files are run from a standard DOS window, they will set the environment but only for that instance of the window. If the window is closed, the batch files would need to be re-run to set the environment variables again. For this reason, the system variables must be set permanently to avoid having to repeatedly run the batch files every time you want to run NONMEM. PDx-Pop also requires "permanent" setting of the environment variables. Permanently setting the environment is done via the Control Panel for the more recent versions of the Windows operating systems (e.g. Windows 2000 or XP). Earlier versions of Windows (e.g Window 95 and 98) used the autoexec.bat and/or the config.sys system files to set the system environment at boot time. (These earlier versions will not be addressed further in this document). Before proceeding with setting the environment variables, it is useful to test the compiler installation to determine the source of the unset variables (the SDK setup or the compiler setup).

Testing the Compiler Installation

The testing consists of three steps that should be done in the following order:

1. Determine if the link.exe program is in the system path to test the SDK setup.
2. Determine if the compiler program is in the system path to test the compiler setup.
   The name of the compiler executable is dependent on the version of the compiler:
   df is used by Digital and Compaq versions
   ifl is used by Intel 7
   ifort is used by Intel 8 and 9
   (fl32 is used by Microsoft Powerstation)

   (The executable names are forward compatible (e.g. ifl will be recognized by Intel 9, but, use of the correct version specific name is recommended to avoid potential version-related issues.)

3. Compile a FORTRAN source file to assure the SDK and compiler are working together. A simple "hello, world" type source file is sufficient and an example, gmtest.for, can be found in the PDx-Pop installation directory.

The following is the expected output from a Windows 2000 system running Compaq Visual Fortran (and Microsoft Visual Studio):

Microsoft Windows 2000 [Version 5.00.2195]

(C) Copyright 1985-2000 Microsoft Corp.

```
C:\>link.exe
Microsoft (R) Incremental Linker Version 6.00.8447
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

usage: LINK [options] [files] [@commandfile]

   options:

      /ALIGN:#
      /BASE:{address|@filename,key}
      /COMMENT:comment
      /DEBUG
      /DEBUGTYPE:{CV|COFF}
…
[more link.exe output here]
…

C:\>df.exe
Compaq Visual Fortran Optimizing Compiler Version 6.6 (Update C)
Copyright 2003 Compaq Computer Corp. All rights reserved.


C:\>cd pdxpop2.0a

C:\pdxpop2.0a>df gmtest.for
Compaq Visual Fortran Optimizing Compiler Version 6.6 (Update C)
Copyright 2003 Compaq Computer Corp. All rights reserved.

gmtest.for
Microsoft (R) Incremental Linker Version 6.00.8447
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

/subsystem:console
/entry:mainCRTStartup
/debugtype:cv
/pdb:none
C:\DOCUME~1\bill\LOCALS~1\Temp\obj300.tmp
dfor.lib
libc.lib
dfconsol.lib
dfport.lib
kernel32.lib
/out:gmtest.exe
```

C:\pdxpop2.0a>gmtest.exe
 Globomax LLC - Tools for Expediting
  Population Analysis (c) 2003

C:\pdxpop2.0a>

The three part test is used because steps 1 and 2 can succeed and 3 will fail (if, for example, a non-compatible SDK version is found by a specific compiler version or if an environment variable, other than the PATH, has not been set).

An example of a failed step 3 might have an output like the following:

C:\pdxpop2.0a>ifort gmtest.for
Intel(R) Fortran Compiler for 32-bit applications, Version 9.1   Build 20060816Z
Package ID: W_FC_C
_9.1.029
Copyright (C) 1985-2006 Intel Corporation. All rights reserved.

Microsoft (R) Incremental Linker Version 8.00.50727.42
Copyright (C) Microsoft Corporation. All rights reserved.

-out:gmtest.exe
-subsystem:console
gmtest.obj
LIBCMT.lib(crt0init.obj) : warning LNK4254: section '.CRT' (40000040) merged into '.data' (C0000040)
 with different attributes

C:\pdxpop2.0a>

Determining which batch file contains the missing environment variables:

The Digital/Compaq/Intel Compilers do not supply the linker program and require the linker to be supplied by Microsoft Visual Studio. However, Digital and Compaq Visual Fortran supply Microsoft Visual Studio as part of their software and automatically installed the Visual Studio components with the compiler components. The batch file supplied by Digital and Compaq to set the environment variables for both the compiler and the SDK is called "dfvars.bat". The file can usually be found in the "C:\program files\DevStudio\DF\bin" directory or in "C:\program files\Microsoft Visual Studio\DF98\bin". As stated earlier, with earlier Digital and Compaq versions, the environment variables for both the compiler and SDK components are often set correctly and you need not be further concerned further with the batch files.

All versions of Intel Fortran require that Microsoft Visual Studio be obtained and installed separately from the compiler. For Intel compilers, there is a batch file supplied to set the Microsoft Visual Studio environment variables and another batch file to set the compiler environment variables.   With Intel (and when the environment is not correctly set for Digital or Compaq) you will have to find the batch files to help you determine the missing environment variables. To further complicate matters, the files have different names depending on the compiler version and the SDK version! The name for the Intel 8 or 9 compiler variables is called "ifortvars.bat". The default location for this file for Intel 9 is "C:\program files\Intel\compiler\fortran\9.1\IA32\bin". The name of the batch file for the recent versions of Microsoft Visual Studio (.Net 2003 and Visual Studio 8) is "sdkvars.bat". The default location for this file for Microsoft Visual Studio 8 is "C:\program files\Microsoft Visual Studio 8\SDK\v2.0\bin". The files have numerous environment variables, but, in general only three critical environment variables need be modified: PATH, LIB and INCLUDE.

The most direct way to determine the environment variables to be completed (or created if not already present) is to study the compiler and/or batch file(s) and then modify the PATH, LIB and INCLUDE variables to include the paths for the compiler and the SDK in these variables.

Another way to determine which component of the environment variables (those due to the SDK or those due to the compiler) is the following algorithm:

1. Create a new directory and copy the sample Fortran source code and the compiler and/or SDK batch files (ifortvars.bat & sdkvars.bat) to the new directory. (This method assumes that the compiler and the SDK have been installed and NONMEM installation may have failed.) In the example to be given Intel 9 with Microsoft Visual Studio 8 have been installed on a Windows XP system.   The directory c:\ctests has been created and gmtest.for, sdkvars.bat and ifortvars.bat have been located and copied to the directory.
2. Open a DOS window and attempt to compile the source file. Was the gmtest.exe file created?
     Yes, then stop. Required variables are defined. Your failed installation was caused by something other than incomplete system variables.
     No, then proceed to step 3.
3. Run the batch file that sets the SDK variables, in this case, sdkvars.bat
4. Again, attempt to compile the source file. Was the gmtest.exe file created?
     Yes, then stop. Required variables are located in the sdkvars.bat file.
     No, then proceed to step 5.

5. Run the batch file that sets the compiler variables, in this case, ifortvars.bat
6. Again, attempt to compile the source file. Was the gmtest.exe file created?
    > Yes. Required variables are located in the sdkvars.bat and/or the ifortvars.bat file. Proceed to step 7.
    > No, stop, your failed installation was caused by something other than incomplete system variables.
7. Once you have determined the components that contain the missing environment variables, examine the batch files, particularly the lines related to LIB, PATH and INCLUDE so that you may add them to your environment.

An example of using the above algorithm:

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\ctests>ifort gmtest.for
Intel(R) Fortran Compiler for 32-bit applications, Version 9.1   Build 20060816Z
Package ID: W_FC_C
_9.1.029
Copyright (C) 1985-2006 Intel Corporation. All rights reserved.

Microsoft (R) Incremental Linker Version 8.00.50727.42
Copyright (C) Microsoft Corporation. All rights reserved.

-out:gmtest.exe
-subsystem:console
gmtest.obj
LIBCMT.lib(crt0init.obj) : warning LNK4254: section '.CRT' (40000040) merged into '.data' (C0000040)
 with different attributes

C:\ctests>dir *.bat
 Volume in drive C is HP_PAVILION
 Volume Serial Number is 6142-03EA

 Directory of C:\ctests

10/09/2006 11:03 AM   967 ifortvars.bat
11/27/2006 02:12 PM   542 sdkvars-sub1.bat
10/02/2006 11:07 AM   847 sdkvars.bat
         3 File(s)   2,356 bytes
         0 Dir(s) 243,543,531,520 bytes free

C:\ctests>sdkvars.bat
Setting environment to use Microsoft .NET Framework v2.0 SDK tools.
For a list of SDK tools, see the 'StartTools.htm' file in the bin folder.

C:\ctests>ifort gmtest.for
Intel(R) Fortran Compiler for 32-bit applications, Version 9.1   Build 20060816Z
Package ID: W_FC_C
_9.1.029
Copyright (C) 1985-2006 Intel Corporation. All rights reserved.

Microsoft (R) Incremental Linker Version 8.00.50727.42
Copyright (C) Microsoft Corporation. All rights reserved.

-out:gmtest.exe
-subsystem:console
gmtest.obj

C:\ctests>gmtest.exe
 Globomax LLC - Tools for Expediting
  Population Analysis (c) 2003

C:\ctests>

In this example, successful compilation of gmtest.for was achieved after
sdkvars.bat was run. The components in sdkvars.bat related to LIB, INCLUDE
and PATH are:

@Set Path=C:\Program Files\Microsoft Visual Studio
8\SDK\v2.0\Bin;C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727;C:\Program
Files\Microsoft Visual Studio 8\VC\bin;C:\Program Files\Microsoft Visual Studio
8\Common7\IDE;C:\Program Files\Microsoft Visual Studio
8\VC\vcpackages;%PATH%

@Set LIB=C:\Program Files\Microsoft Visual Studio 8\VC\lib;C:\Program
Files\Microsoft Visual Studio 8\SDK\v2.0\Lib;%LIB%,

@Set INCLUDE=C:\Program Files\Microsoft Visual Studio
8\VC\include;C:\Program Files\Microsoft Visual Studio
8\SDK\v2.0\include;%INCLUDE%

The next step is to add the path within the above statements to the PATH, LIB
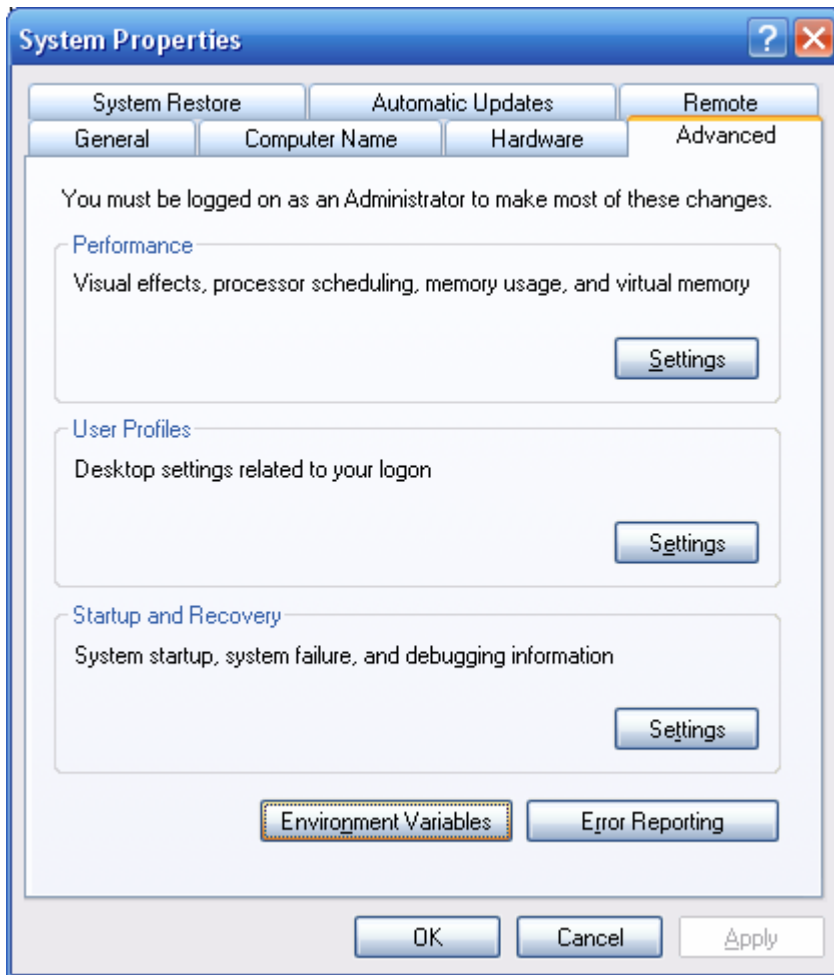and INCLUDE variables using the Control Panel:
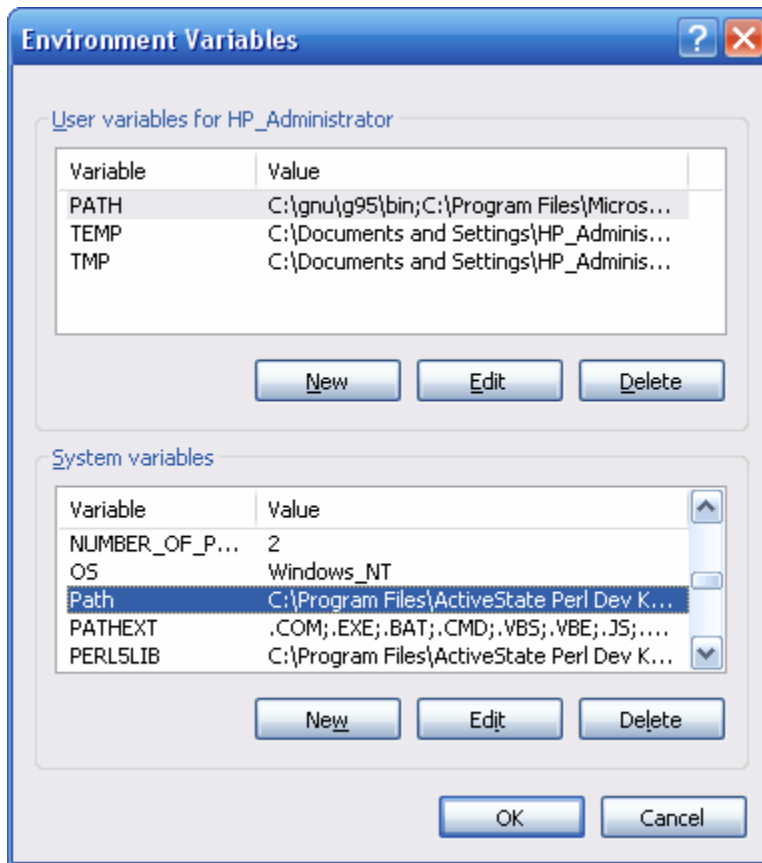
Control Panel > System > Advanced > Environment Variables.

Then select and edit each of the variables from the System Variables (not the User Variables) to include the appropriate paths.
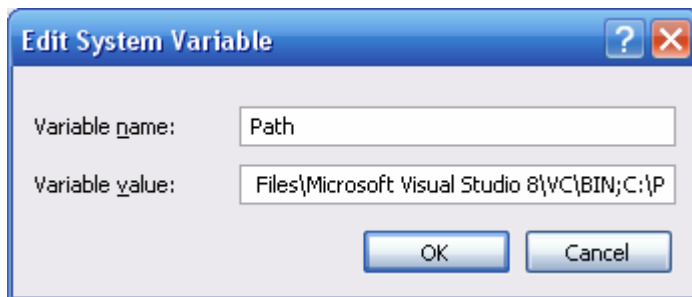
At the end of the variable value, additional paths can be added (separated by semicolons).

Finally, after all three variables have been modified, close the Control Panel and open a new DOS window to test compiling the source file. If successful, you can now proceed with the NONMEM installation.

INSTALLING THE MINGW G95 FORTRAN COMPILER

The g95-MinGW.exe g95 installer extracts the files needed for g95 to run on supported Windows systems with or without MinGW already installed. Where MinGW is not installed a minimal set of support files from the MinGW distribution and the MinGW binutils package is required. These can also be installed if installing into MinGW. Doing this, especially if doing a first time g95 install, will ensure that all the files that g95 needs to run are available immediately. To do this, when the installer pops up a message box asking: "Install MinGW Utilities and libs?", respond by clicking "Ok".

The PATH environment variable may then be set to include the path to MinGW and the g95 executable. I recommend using the path "C:\gnu\g95" for your installation.

- Avoid directory names containing spaces, e.g. "c:\Program Files\.."
- Avoid duplicate entries in your PATH.
- Set the LIBRARY_PATH environment variable to the location of your MinGW\lib directory (C:\gnu\g95\lib if above suggested directory was used.)

In Windows NT/2000/XP the PATH variable can be set by going to the Control Panel, selecting System, and then locating the environment (or advanced, environment) section.

Add a new variable named LIBRARY_PATH and set its value to <MinGW_library_install_path>, i.e., the full path to the MinGW\lib directory.

Similarly add a new variable named PATH (or edit it if already present) and set its value to:

   <MinGW_install_path>;<g95_install_path>;%PATH%

 The cdsetup6.bat file for installation of NONMEM VI is compatible with g95. Use "g95" as the compiler argument for this compiler with cdsetup6.bat. The NONMEM installer included with PDx-Pop 2.x is also compatible with g95. [In both cases, the optimization used with g95 is "-O -fsloppy-char".] If necessary, a version of cdsetup.bat for installation of NONMEM V with g95 can be provided.

Installing Open Watcom 1.5:

Obtain at http://www.openwatcom.org/index.php/Download

(actual download occurs from http://openwatcom.mirrors.pair.com/)