Some Notes on Recompiling TRNSYS as a Platform Independent Executable Program

At its core, TRNSYS has always been a FORTRAN program. All of the standard component models are written in FORTRAN77 with a few exceptions written in FORTRAN90. With the rise in popularity of the Windows operating system during the 90s, a number of operating system based features (such as the online plotter and communication with external programs) were added to increase the program's capabilities. However, it should be pointed out that by shutting off these options, TRNSYS can be recompiled as a FORTRAN Console Application (a platform independent executable.) This document describes a procedure for recompiling and running TRNSYS in this manner. It deals with one compiler; the Compaq Visual Fortran developer studio version 6.1. Fortunately, the Compaq compiler is built on the same principles as a number of other compilers so it should be relatively easy to take these directions and apply them to other compilers. Other similar compilers include Microsoft's Powerstation 4.0, and Digital's Visual Fortran versions 5 and 6. In reality, any Fortran compiler can be used to recompile TRNSYS. At the Solar Lab, we have experience with the Compaq compiler encourage its use. Please look at our web page (http://sel.me.wisc.edu/trnsys) for the most up to date information about using Fortran compilers with TRNSYS.

Creating the Executable

1. Under the *File* main menu, select *New*. A New dialog box will appear with the *Project* tab selected. From the list of projects in this window, select *Fortran Console Application*. Give the project a name and a directory (for example, *trnexe* in the *c:\trnsys15* directory). Once selected, a dialog will come up asking for the type of Application that you want to create. Select *Empty Project*.

2. Under the *Project* main menu, select *Add to Project*. From this side menu, select *Files*. Add all of the Fortran files (*.for*) from the *Kernal* and *Types* subdirectories in the */trnsys15* directory with the exception of *\trnsys15\kernal\trnsys.for*, *\trnsys15\kernal\exec.for* and *\trnsys15\kernal\clock.for*. You will also need to add certain files from the *\trnsys15\kernal\OS_comp* directory. Add *trnsys_exe.for*, *clock_exe.for* and *exec_exe.for*. NOTE: if you are using a different compiler/operating system combination, replace the file *\trnsys15\kernal\trnsys_win32_cvf90.for* with the file appropriate to your combination. The replacement files should be found in the *\trnsys15\kernal\OS_Comp* directory. Finally, add the file *dvfdummy.lib* from the *Dummy* subdirectory. Add any non-standard types that you wish to use in a TRNSYS simulation.

4. There are a few Types that need to be removed because the make specific use of the Windows operating system. These are: online.for, Type61.for and Type66.for. Select each one in the window at the left of your screen and press the delete key.

5. Under the *Project* main menu, select *Settings*. There are numerous settings here controlled by a series of tabs at the top. Below are the important settings for each of these tabs. Each tab can have several "Categories" so there are many settings involved. Only the settings that must be changed from their default values are mentioned.

6. In the *Settings For* pull-down menu, select *Win32 Debug* so that you are modifying the settings for the Debug version.

Debug Tab
General
Executable for session: c:\trnsys15\trnexe.exe
Working Directory: c:\trnsys15
Additional DLLs
Uncheck *Try to locate other dll's*

Fortran Tab
General
Click the box for "Generate Source Browser Information
Libraries
Use Fortran Run-time Libraries should be set to "Static *"
Preprocessor
Enter \trnsys15\include" for *Custom INCLUDE and USE paths*
Run Time
Click box by all "Runtime Error Checking" options
Click box by *Generate Traceback Info*

Link Tab
General
Output File Name: c:\trnsys15\trnexe.exe
Check box: Generate Debug info
Customize
Output filename should now read c:\trnsys15\trnexe.exe
Input
Under Ignore Libraries: enter "libcd.lib" *

* NOTE: some users of CVF6.1 reported having to ignore a library called libcmt.lib instead of libcd.lib.

7. Under the *Build* menu, click *Rebuild All*. Watch the bottom window for errors and problems with compiling and creating the trnlib.dll output file. If everything worked correctly, the TRNSYS executable was created in the *\trnsys15* directory and TRNSYS can be run on any platform. To run TRNSYS, select *Execute* from the *Build* menu. TRNSYS will prompt you for a TRNSYS input file. If you are not interested in running TRNSYS from the developer studio, you can simply double click the trnexe.exe icon that should now be in your \trnsys15 directory.

NOTE ***NOTE*** ***NOTE*** ***NOTE*** ***NOTE*** ***NOTE***
When you installed TRNSYS originally, the Windows operating system associated the .dck file extension with the TRNSHELL program. Because of this, when you enter the deck name that you wish to run into the TRNSYS console window, TRNSHELL will be automatically launched and your simulation will not run. To get around the problem, it may be necessary to hardwire the deckname into the trnsys.for code.

      a. open trnsys.for and comment out the line: `READ(LUK,12) PNAME`
      b. comment in the line: `PNAME = 'path to deck'`
      c. replace `'path to deck'` with the path to the dck file that you wish to run. For example `'c:\trnsys15\fixed.dck'`

Using the Developer Studio Debugging Features

1. To use the CVF61 debugging features, select *Start Debug* from the *Build* menu, then *Go*. At this point, the program will state "TRNSYS.EXE does not have any debug information. OK?" Click OK because we are concerned with the Fortran DLL and not the TRNSYS executable. Click the box by *Don' t ask this in the future* and click OK. Without a breakpoint or a bug, TRNSYS will just start and run until the simulation is finished.

2. To halt execution at some point in the source code, insert breakpoints at one or more lines in the Fortran code of one or more subroutines. For example, open the file *trnsys.for* and create a breakpoint at the beginning of the source code, perhaps in the definitions of variables and common blocks. To create a breakpoint, put the cursor on a line of the source code and press F9. A red dot will appear to the left of the source code line. When TRNSYS is executing through the Debug feature of DVF, it will always stop at this point in the code.

3. Experiment with the other debug commands by setting (and turning off) breakpoints with the F9 key, adding watch variables, stepping through the source code (F11), and running until you reach the cursor (F7). You can see the value of a variable simply by placing your cursor on top of that variable in the source code for a couple of seconds.

4. You should modify the default way in which the debugger handles exceptions (for example, an array-out-of-bounds error). When TRNSYS halts at the red dot, select the *Debug* main menu and then the item *Exceptions*. Note that the *Debug* main menu is not visible unless the debugger is running. In the *Exceptions* dialog box, select all the exceptions from the list using the Shift key. Then, select *Stop Always* and click *Change*. This is important so that the debugger will stop and display the line that has a run-time error such as Array out of Bounds.

5. Under the *Tools* main menu, select *Options*. Choose *Tabs* and then select *Insert Spaces* if it says *Keep Tabs*. This can save a lot of hassle if you edit in CVF61 and then open the file with another editor that does not accept tabs.

6. Familiarize yourself with the *Find in Files* option under the *Edit* main menu. This feature allows you to search for text strings throughout all the Fortran code in your project. This is very convenient if you are adjusting COMMON blocks in many subroutines, for example. 12. Familiarize yourself with the various commands under the *Debug* main menu. Also note that it is possible to have Conditional Breakpoints by selecting *Breakpoints* from the *Edit* main menu.