Using Common Fortran Compilers with TRNSYS

This document describes how to add the component that you just created to the trnlib.dll, which contains all the TRNSYS source code. The information is also available at the end of the third chapter in the main TRNSYS manual (Volume 1). This document deals with two compilers; the Microsoft PowerStation FORTRAN Developer Studio and the Digital Visual Fortran developer studio. In reality, any Fortran compiler that can create a 32 bit Dynamic Link Library (.dll) can be used to recompile TRNSYS. At the solar lab, we have experience with these two developer studios and encourage their use. Please look at out web page for the most up to date information about using Fortran compilers with TRNSYS.

Digital Visual Fortran 5.0 (DVF50), 6.0 (DVF60) and Compaq Visual Fortran 6.1 (CVF61)

From the Command Line

The following command line (DOS prompt) statement will allow the compiler to compile a Fortran subroutine in preparation for linking:

fl32 /c /I\trnsys15\include <filename.for>

To compile a subroutine using the *Compile Module* menu item in the TRNSHELL program, enter the *File/Setup/Compiler* menu and enter the line above on the *Compile* line (the path for the compiling executable will depend on your installation of DVF50, DVF60 or CVF61).

In order for TRNSYS to use a subroutine, it needs to be included in the Dynamic Link Library TRNLIB.DLL. The file *link.dvf*, (or link.cvf if you are using Compaq's Visual Fortran) contained in the *\trnsys15* directory of the TRNSYS installation, is a link file for DVF50 or DVF60 containing all of the standard TRNSYS components. Add any non-standard components to *link.dvf* by typing the command:

edit link.dvf

Refer to the existing *link.dvf* file for proper syntax. If you are using CVF61 compiler, edit *link.cvf* instead of *link.dvf*

Once all components have been added to *link.dvf* (or *link.cvf*), TRNLIB.DLL can be created. The
following statement will link the Fortran object code listed in the file *link.dvf* (or *link.cvf*) to form the Dynamic Link Library TRNLIB.DLL:

link @link.dvf
(or *link @link.cvf*)

Also included with the TRNSYS package is a batch file called *alltrndf.bat*. Running this batch file will compile all of the Fortran code in the *Kernal* and *Types* directories and link them (along with *dvfdummy.lib* from the *Dummy* directory) to make TRNLIB.DLL in the *\trnsys15* directory using CVF61. The *Dummy* routines are stored in the library *dvfdummy.lib* and are only called if the corresponding TYPE routine is not linked in the TRNSYS DLL. To run *alltrndf.bat* using the *Rebuild TRNSYS* menu item in the TRNSHELL program, enter the *File/Setup/Compiler* menu and enter *\TRNSYS15\ALLTRNDF.BAT* on the *Rebuild* line.

From the Developer Studio

1. If you are using DVF50: 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 *Win32 Dynamic Link Library*. Give the project a name and a directory (for example, *dvftrnwin* in the *c:\dvftrnsys* directory).

If you are using DVF60 or CVF61: 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 Dynamic Link Library*. Give the project a name and a directory (for example, *dvftrnsys* in the *c:\dvftrnsys* directory). Once selected, a dialog will come up asking for the type of DLL that you want to create. Select *Empty dll Application.*

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. You will also need to find the files called Type56.obj and Type56_mod.obj that correspond to your Fortran Compiler. In the *Types* directory, locate the folder called Type56 and open it. Locate the folder that corresponds to your Fortran compiler, open it and Add Type56.obj and Type56_mod.obj to the project. If you add Type61.for to your project, you will also need to add a library: \trnsys15\include\extdll.lib to your project. Finally, add the file *dvfdummy.lib* from the *Dummy* subdirectory. Add any non-standard types that you wish to use in a TRNSYS simulation.

3. 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.

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

If you're using DVF 5.0…

If you're using DVF 6.0 or CVF6.1…

Debug Tab

General

Executable for session: c:\trnsys15\trnsys.exe

Working Directory: c:\trnsys15

Program Arguments: Here you can enter a TRNSYS input file name

Additional DLLs

Local Name: c:\trnsys15\trnlib.dll (note: if you select both items in the left "Settings for:" box, you cannot alter this category. You must select just the *Debug* project).

Debug Tab

General

Executable for session: c:\trnsys15\trnsys.exe

Working Directory: c:\trnsys15

Program Arguments: Here you can enter a TRNSYS input file name

Additional DLLs

Local Name: c:\trnsys15\trnlib.dll (note: if you select both items in the left "Settings for:" box, you CAN now alter this category).

	Uncheck *Try to locate other dll’s*
Fortran Tab	Fortran Tab
General	*General*
Click the box for “Generate Source Browser Information	Click the box for “Generate Source Browser Information
Compatability	
Under “Powerstation 4.0 Compatability Options” check the box labeled *General*	
	Libraries
	Use Fortran Run-time Libraries should be set to “Static *”
Preprocessor	*Preprocessor*
Enter \trnsys15\include” for *Custom INCLUDE and USE paths*	Enter \trnsys15\include” for *Custom INCLUDE and USE paths*
Run Time	*Run Time*
Click box by all Extended Error Checking options	Click box by all “Runtime Error Checking” options
	Click box by *Generate Traceback Info*
Link Tab	Link Tab
General	*General*
Output File Name: c:\trnsys15\trnlib.dll	Output File Name: c:\trnsys15\trnlib.dll
Check box: Generate Debug info	Check box: Generate Debug info
Customize	*Customize*
Output filename should now read c:\trnsys15\trnlib.dll	Output filename should now read c:\trnsys15\trnlib.dll
Input	*Input*
Under Ignore Libraries: enter “libcd.lib”	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.

5. 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 DLL was created in the *\trnsys15* directory and TRNSYS can be run with the updated DLL. To run TRNSYS, select *Execute* from the *Build* menu. TRNSYS will prompt you for a TRNSYS input file.

6. To use the debugging features of DVF50, DVF60 or CVF61, select *Start Debug* from the *Build* menu, then *Go*. At this point, the program will state that "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.

7. 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.

8. 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.

9. 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.

10. 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 DVF50 or DVF60 and then open the file with another editor that does not accept tabs.

11. 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.

Microsoft FORTRAN PowerStation 4.0 (MSFPS40)

From the Command Line

The following command line (DOS prompt) statement will allow the compiler to compile a Fortran subroutine in preparation for linking:

```
fl32.exe /c /Ox /G5 <filename>
```

To compile a subroutine using the *Compile Module* menu item in the TRNSHELL program, enter the *File/Setup/Compiler* menu and enter *\MSDEV\BIN\FL32.EXE /c /Ox /G5* on the *Compile* line (the path will depend on your installation of MSFPS40).

The following statement will link the Fortran object code listed in the file *link.fp4* to form the Dynamic Link Library TRNLIB.DLL:

link @link.fp4

The file *link.fp4*, contained in the *\trnsys15* directory of the TRNSYS installation, is a link file for MSFPS40 containing all of the standard TRNSYS components. Refer to this file for the proper syntax of the MSFPS40 link file.

Also included with the TRNSYS package is a batch file called *alltrn32.bat*. Running this batch file will compile all of the Fortran code in the *Kernal* and *Types* directories and link them (along with *dumf40.lib* from the *Dummy* directory) to make TRNLIB.DLL in the *\trnsys15* directory using MSFPS40. The *Dummy* routines are stored in the library *dumf40.lib* and are only called if the corresponding TYPE routine is not linked in the TRNSYS DLL. To run *alltrn32.bat* using the *Rebuild TRNSYS* menu item in the TRNSHELL program, enter the *File/Setup/Compiler* menu and enter *\TRNSYS15\ALLTRN32.BAT* on the *Rebuild* line.

Another batch file, *maktrn32.bat*, included in the TRNSYS package, allows the user several options for creating a new TRNSYS DLL depending on whether changes have occurred in the source code, input file, or both. The user can save time by having *maktrn32.bat* check only for source code or input file changes. If the user indicates that a change in source code has occurred, the *maktrn32.bat* script compares the creation date of the TRNSYS DLL with the last modification date of the TRNSYS source code files and compiles and links the subroutines if necessary. The link file *link.fp4* is used by *maktrn32.bat*. If the user indicates that a different or changed input file is being used, the *maktrn32.bat* script will search through the input file for the necessary types and create a new *link.fp4* file. If using the TRNSHELL program, the user may wish to use *maktrn32.bat* as the *Rebuild TRNSYS* command. The *maktrn32.bat* utility can not compile new source code the very first time. That is, a version of the object code (*.obj*) must exist (in the *\trnsys15\types* directory) before the *maktrn32.bat* can perform the update. Also, if a new component (*.for*) is written and placed outside the *\trnsys15\types* subdirectory, *maktrn32.bat* will be unable to automatically compile this routine.

From the Developer Studio

Follow these instructions to get started using the MSFPS40 Developer Studio for handling the compiling, linking, and debugging of Fortran subroutines for TRNSYS.

1. Start the MSFPS40 Developer Studio. Under the *File* menu, create a new *Project Workspace*. From the list of "Types of Projects" select Dynamic Link Library. Give the

project a name. For example, you might name the project *32trnsys* and place it in the *c:\32trnsys* directory.

2. Under the *Insert* menu, click *Add Files to Project*. Add all of the Fortran files (*.for*) from the *Kernal* and *Types* subdirectories in the */trnsys15* directory. Add the file *dumf40.lib* from the dummy subdirectory. Add your own custom components, if necessary.

3. Under the *Build* menu, click *Settings*. You will notice that there are numerous settings here controlled by a series of tabs. Below are the important settings for each of these tabs. Each tab can have several submenus, so there are many settings involved. Only the settings that must be changed from default values are described below.

First, there should be two items in the left box called (in this example): *32TRN - Win32 Debug* and *32TRN - Win32 Release*. Be sure you have the *Debug* version selected.

Debug Tab
General
Executable for session: c:\trnsys15\trnsys.exe
Working Directory: c:\trnwin
Program Arguments: 2>\32trnsys\error.txt
Additional DLL's
Local Name: c:\trnsys15\trnlib.dll (note: if you select both items in the left "Settings for:"box, you cannot alter this category. You must select just the Debug project).

Fortran Tab
Fortran Language
Click box by Extended Bounds and Error Checking

Link Tab
General
Output File: c:\trnsys15\trnlib.dll
Object/library modules: blank
Check box "Generate debug info."
Customize
Output filename should now read c:\trnsys15\trnlib.dll

4. Under the *Build* menu, click *Rebuild All*. Watch the bottom window for errors and problems with compiling and creating the trnlib32.dll output file. If everything worked correctly, the TRNSYS DLL was created in the *\trnsys15* directory and TRNSYS can be run with the updated DLL. To run TRNSYS, select *Execute* from the *Build* menu. TRNSYS will prompt you for a TRNSYS input file.

5. To use the debugging features of MSFPS40, select *Debug* from the *Build* menu, then *Go*. At this point, the program will state that "TRNSYS.EXE does not have any debug information. OK?" Click OK because we are concerned with the Fortran DLL and not the

TRNSYS executable. Without a breakpoint or a bug, TRNSYS will just start and run until the simulation is finished.

6. 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 click F9. A red dot will appear to the left of the source code line. When TRNSYS is executing through the Debug feature of MSFPS40, it will always stop at this point in the code.

7. Experiment with the other debug commands by setting (and turning off) breakpoints with the F9 key, adding watch variables, stepping through the source code (F8), 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.

8. If your source code has a serious bug, such as trying to access an array outside of its boundaries, the debugger will simply stop and say that TRNSYS has exited with Code 1 (or something like that). At this point, access the file c:\32trnsys\error.txt. In our setup (the *Debug* tab), we have redirected the error output to this file since TRNSYS.EXE will not allow the debugger to write the error message (the line number of the error, for example) to the screen.

Microsoft Fortran 5.1 (MSF51)

The MSF51 compiler is a 16-bit compiler. This compiler cannot be used to create the 32-bit Dynamic Link Library used by the TRNSYS executable. The TRNSYS package is no longer supported for 16-bit environments. If you wish to use the MSF51 compiler with the discontinued 16-bit version of the TRNSYS executable or for a non-Windows application of TRNSYS (linking the source code to form an executable), the information below may be useful. The following command line (DOS prompt) statement will allow the compiler to compile a Fortran subroutine in preparation for linking:

```
FL.EXE /c /FPi /AL /Aw /Gw /Gt <filename>
```

The following statement will link the Fortran object code listed in the file *link.trn* to form the Dynamic Link Library TRNLIB.DLL for use with the discontinued 16-bit version of the TRNSYS executable:

```
link @link.trn
```