CERES Software Bulletin 95-06

F90 Debugger General Information, July 14, 1995

Sandy Nolan (s.k.nolan@larc.nasa.gov)

Purpose:

There have been several simultaneous efforts by CERES Data Management Team members to use the dbx utility to debug programs written in Fortran 90. This Bulletin contains information gathered from several of these efforts.

Information:

Georgia Liu has solved the initial F90 debugger problem. Her tracking down the solution was not a trivial effort and required much interaction with the vendor.

After making sure that we had compatible versions of dbx and F90, Georgia found out that the trick to using NAG F90 and dbx is to use the "c" name for all variables and subroutines in dbx (because dbx is actually using the "C" version of the F90 program). In most cases, a "_" needs to be added to all names. If you have a subroutine named suba and a variable named clock, then in dbx, you must refer to the subroutine and variable as suba_ and clock_.

To check on the "c" version of subroutine and variable names, we compiled F90 source code with the -S option. This produced the "C" code used in compilation and contained the names recognized by dbx. (The "c" files will have the same names as the f90 source code but will have .c extensions. If you are using the makemake utility to create your Makefile, be sure to move these .c files to another directory before executing makemake. Also change the -O and -s flag options in your Makefile to -g, before making your executable.)

I was able to use both dbx and dbxtool (the window version of dbx) to debug with my (non- Toolkit version) F90 code. I was also able to use dbx to debug F90 code which included calls to the PGS Toolkit, but could not use dbxtool with F90 code using the Toolkit. Ed Howerton and Sunny Sun-Mack are also looking at the problem of trying to use dbxtool with F90 code that uses the Toolkit.

I also tried using "include files" with dbx. When the include files contained non-executable code, the line numbers and steps through the program were all correct. When the include files did contain executable code, dbx would not step into those lines of code in the include file, but would step over them as it executed them (as the "next" dbx command would step over (i.e. not step into) a subroutine or function call.) In both cases, the line numbers in dbx were correct.

For further information please see file, F90-Users-Notes, which is attached to this Bulletin.