

## CCP4 and Fortran 90

Here follows some random observations on Fortran 90. They are not particularly astute, but are made public in the hope of stimulating some thought in this area.

---

The CCP4 suite currently uses Fortran 77 with a few obsolescent features, and a few Fortran 90 extensions which are generally accepted by f77 compilers (IMPLICIT NONE, END DO, long names, etc.).

But Fortran has moved on with Fortran 90 well-established, and even rumours of a Fortran 95 ;-). Fortran 90 includes new control structures (e.g. SELECT CASE), user-defined data types (cf. structs in C), whole array notation ( $a = b + c$  where a,b,c are arrays), modules (MODULE, USE), dynamic memory allocation (ALLOCATE, DEALLOCATE) plus more. Wouldn't it be nice to use these extra features?

Well, yes, but while there are many Fortran 90 compilers out there, they are not installed on all systems. In particular, there is no GNU g90 (although Pacific-Sierra Research offer a Linux VAST/f90 Fortran 90 compiler (<http://www.psrv.com/lxf90.html>) free for personal use, and Lahey (<http://www.lahey.com/>) apparently supply a command line version for free). Distributing Fortran 90 code would therefore cause problems for many users. It should also be mentioned that compiled Fortran 90 code is reported to be a little slower than the Fortran 77 equivalent, although this of course depends on the particular code and compiler.

Anyway, the first step in migrating to Fortran 90 is testing that the current code compiles under f90. Since Fortran 90 is supposed to be backwardly-compatible to Fortran 77 it should. Here's the first test ....

## CCP4 with IRIX f90 compiler

We have the MIPSpro Fortran 90 compiler Version 7.2.1 on IRIX6.5 Note: compiler options seem to be migrating to the option group syntax, e.g. `-Olimit 3000` has to be replaced by `-OPT:Olimit=3000`, `-trapuv` is to be replaced by `-DEBUG:trap_uninitialized=ON`, etc.

Compiling the library brought up 2 problems:

1. In `unix.m4`, the `READONLY`, `CARRIAGECONTROL=CCNTRL` and `DISPOSE=DISP` specifiers for `OPEN` were not allowed. These were simply left out.
2. In `unix.m4` and `ccplib.f`, the `BYTE` data type was not allowed. The compiler was happy with `INTEGER*1` as a replacement, but I'm not sure if this is entirely equivalent? Supposedly `INTEGER*1` for signed byte, and `LOGICAL*1` for unsigned byte. To be tested .... (Should really specify `KIND` value, but then f77 compiler would complain.)

With the library compiled, there were no problems compiling the programs in `$CCP4/src` and `$CCP4/unsupported/src` Well actually, f90 fell over on `getax.f`, `dmmulti_/dmmulti_dmaver.f`, `refmac_/lsq.f`, `sftools_/sftools.f` but these were due to coding mistakes rather than language problems.

At runtime, the programs gave out incomprehensible warnings, which seem to arise from the `-trapuv` a.k.a. `-DEBUG:trap_uninitialized=ON` compiler option. Otherwise, the programs I tried gave the same results as those compiled by f77.

# Real Fortran 90 with IRIX f90 compiler

A random list of observations:

1. The switch `-freeform` is required to allow free form code layout.
2. Whole array processing seems to work, and is cute!

## Fortran 90 extensions available in g77

Some Fortran 90 features are actually available in g77, though of course these are just the simple features to do with allowable syntax, rather than those giving extra functionality (e.g. whole array notation).

Specifically, with g77 version 0.5.21 I found:

### Allowed

- Same-line comments beginning with `!`
- `END program main, etc.`
- More than one statement per line separated by `;`
- `integer :: i=5, etc.`
- `ENDDO, CYCLE, EXIT` for DO loops
- `SELECT CASE`

### Not allowed

- Free source form, unless `-ffree-form` specified
- Continuation character `&`
- `real, parameter :: pi=3.14 or integer, dimension (10) :: i`
- Whole array notation
- `USE, MODULE`
- Derived data types, `TYPE`

---

*Martyn Winn: m.d.winn@dl.ac.uk*

Last modified: Fri Nov 20 15:02:05 GMT 1998