

# Re: Fortran module checking

---

*Source:* <http://coding.derkeiler.com/Archive/Fortran/comp.lang.fortran/2007-06/msg00200.html>

---

- *From:* Ken Fairfield <[Ken@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Ken@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Wed, 06 Jun 2007 19:40:18 -0700
- 

Eigenvector wrote:

I'm a serious newbie to Fortran and am asking this because I'm being forced to help debug another unwilling person's code. Forgive the lack of understanding.

The code that I'm interested in uses modules to retrieve variable names for the main code. Essentially there is a `variable.f` containing all the code variables which is turned into a `.mod` file – that is then called by `main.f` with a `USE` statement.

It would appear through my debugger (idebug) that attempts to call code in that `.mod` file are returning segmentation faults. But I can't prove that and the developer is balking. So I'm looking for ways to demonstrate that the contents of the `.mod` file are being included in the finished product and available to the executable – that the syntax is proper. He's a developer and believes my system is in error, I'm a Sys Ad and believe his code is in error so I'm trying to establish the validity of my position by demonstrating that either I'm right or wrong.

Is it as simple as (mind you I'm using xlf90)

```
! variable.f psuedocode  
variable1 = tag1, tag2
```

xlf90 variable.f -o variable.mod (I'm paraphrasing here)

```
! main.f psuedocode  
use tag1="Hello"  
use tag2="Hi"
```

xlf90 main.f -o program.exe

Or are there compile options that should be invoked to ensure that `USE` statements get included into the main code?

In addition to the comments of Beliaevsky and dpb, note that compiling a source file containing a `MODULE` produces *both* a module output file (typically `.mod`) *and* an object file (`.o` or `.obj`, etc.).

Compiling a source file that `USES` a module requires the `..mod` file be present/accessible. Linking the program

## Re: Fortran module checking

requires the .o/.obj files from all USEing files \*plus\*  
all USEd modules.

Finally, and to reinforce the comments of the others,  
\*none\* of what you've shown is Fortran so it's basically  
impossible to diagnose the error(s). Please post the  
actual source code involved, or at the very least, the  
declarations in the module, and the USE statements and  
references to the module variables and/or routines in  
the main program...exact cut'n'paste please, no paraphrasing...

–Ken

—

Ken & Ann Fairfield

What: Ken dot And dot Ann

Where: Gmail dot Com

.