

## Re: Reference to derived type element by index?

---

*Source:* <http://coding.derkeiler.com/Archive/Fortran/comp.lang.fortran/2008-03/msg00836.html>

---

- *From:* "James Van Buskirk" <[not\\_valid@xxxxxxxxxxxx](mailto:not_valid@xxxxxxxxxxxx)>
  - *Date:* Sun, 30 Mar 2008 12:20:13 -0600
- 

"Rob Crain" <[r.a.crain@xxxxxxxxxxxx](mailto:r.a.crain@xxxxxxxxxxxx)> wrote in message  
[news:fsokhs\\$kdas1@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:fsokhs$kdas1@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

Since it's not mission critical and I'm pushed for time I opted for an inelegant select case statement; it does the job. Thanks to all for your responses though – when I get time to tune the code I'll revisit Dick's suggestion, it looks like it might be the answer.

Since C binding was introduced, you can point at any TARGET with any kind of pointer you want, just like in C.

```
C:\gfortran\clf\fake_label>type fake_label.f90
module fake_label_mod
implicit none
private
integer, parameter :: circle_label_len = 20
public circle_datatype
type circle_datatype
real radius
real x
real y
real z
character(len=circle_label_len) circle_label
end type circle_datatype
end module fake_label_mod

program fake_label
use fake_label_mod
use ISO_C_BINDING
implicit none
type(circle_datatype), target :: circle1
real, pointer :: pcircle1(:)
integer i

call C_F_POINTER(C_LOC(circle1), pcircle1, [4])
pcircle1 = [(i,i=3,15,4)]
circle1%circle_label = "It's a circle"
```

Re: Reference to derived type element by index?

```
write(*,*) circle1  
end program fake_label
```

```
C:\gfortran\clf\fake_label>c:\gfortran\win64\bin\x86_64-pc-mingw32-gfortran  
fake  
_label.f90 -ofake_label
```

```
C:\gfortran\clf\fake_label>fake_label  
3.0000000 7.0000000 11.000000 15.000000 It's a  
circle
```

Thus in the above I have pointed at the four REAL elements of circle1 with a pointer to an array of REALs, and accessed them as an array of REALs.

```
---  
write(*,*) transfer((/17.392111325966148d0,6.5794487871554595D-85, &  
6.0134700243160014d-154/),('x'/)); end
```

.