Subject: Re: declare variables
Posted by James Kuyper on Tue, 18 Oct 2005 21:27:40 GMT
View Forum Message <> Reply to Message

qian wrote:
> Hi,
>
> I am using FORTRAN and IDL at the same time, so sometime I just assume
> some FORTRAN rules when using IDL. I just find out that the variable
> type can change within an IDL program, even you declare it to be some
> specific type. For example:

You don't declare an IDL variable to have a given type. Whenever you assign a value to an IDL variable, that variable automatically acquires the data type of whatever it is you assigned into it.

> data=dblarr(2)

This isn't a declaration; it's an assignment statement like any other. Since dblarr(2) is a 2-element array of doubles, that's the type of 'data'.

> data=[2.3, 3.4]

Since [2.3, 3.4] is a 2-element array of floats, that's the new type of 'data'.

Note: if you type data[0] = 2.3, then 2.3 would have been converted to a double before storing it in data[0]. A variable's type only changes when you assign a value to the entire variable, not when you assign to a single element of an array.

...
> But if I read in data from other files, in which '1' actually means
> '1.0', is it possible I can re-inforce the array to be certain type (or
> let the variables to be certain type throughtout the whole program)? So
> even if I type d=1, it is still a real number?

You don't have to worry about that; it doesn't work that way. When you're reading data from a text file using commands like:

```
READF, infile, myarray
```

Then myarray must already exist, which means it already has a specific type, size, and array shape. The interpretatation of the text file is controlled by the data type and size of myarray. If myarray is currently an array of 26 integers, the above command will try to interpret the file as containing 26 integers (and it will complain if

that interpretation doesn't work). If myarray is a 4x4 array of double precision numbers, it will attempt to interpret the file as containing 16 floating point numbers.