Subject: Re: Grumbling about setting double precision
Posted by Karl[1] on Thu, 22 Dec 2011 20:08:23 GMT
View Forum Message <> Reply to Message

Editing: A lot of times, constant arrays like this come from someplace else and someone wrote a script or program to generate them. Such a script or program can be modified to add the 'd'. If that is not feasible, there are plenty of editors that support keyboard macros which can be run repeatedly for a given number of times. Still tedious but not as bad.

Regular expressions are also possible. In Notepad++:

Find: ([0-9]*[\.]*[0-9]*)([,\]])
Replace: \1d\2

Press Replace All and you're done. If that make you nervous, you can restrict the range of Replace All by selecting the text and checking the "In selection" box.

Memory: It is probably a mistake to default to double. It would take up more space and cause other problems. Variables that are created as a result of this array being part of the calculation would be promoted unnoticed to double as well, increasing memory usage even more. And the change in precision may affect results.

I do like the

b=[24.5d,9999.9]

approach. It expresses intent and doesn't seem that confusing (to me). It isn't perfectly backwards-compatible, but if someone wrote the above, they probably wanted a double array anyway.

It should be easy to do in IDL. IDL needs to analyze the first array element in order to decide what sort of array to make; int, string, or float. It would surely know if the literal is double or float at that point.