Subject: Re: Bizarre (?) behavior of randomu
Posted by M. Katz on Wed, 02 Jan 2008 22:24:07 GMT
View Forum Message <> Reply to Message

The trick is to preserve the 'seed' from one run to the next.
If you put the seed in a common block, for example, then it won't keep
getting reset every time you run the program.

So the first two lines of your program would become

```
pro rannummers
  common rblock, seed  ;--- common blocks need to be named.
```

I'm not an expert in (pseudo) random number generation, but I believe
that the whole point of the 'seed' argument is for when people need to
re-generate the exact same sequence of random numbers. You can see by
a simple test that if you give randomu() the same seed value, you get
the same sequence of numbers--every time.

Consequently, every time you restart IDL and run this program, you'll
get the same sequence of random numbers. That's the expected behavior.
Since you want different results every time, you have to set the seed
to a different value each time. For that, you could use the system
clock {systime(1)}, or some such thing you choose. Preserve the seed
using the common block, as above, and then you're all set.

Usually common blocks are frowned upon as a programming technique.
It's a sneaky way of passing information around that bypasses
arguments and keywords. Yet in this case, I think it's justifiable.

M.