Subject: screwy system variables !? [2]
Posted by trujillo on Thu, 25 Jul 1996 07:00:00 GMT
View Forum Message <> Reply to Message

(Had some posting problems on the first one)
It seems like I can't actually create new system variables in a
function or program and use them later in that same function/program.
For example, I have the following program (in file tester.pro):
> pro tester
> defsysv, "!tester", 222
> print, !tester ;this line directly refers to !tester
> end

When I try to compile/run it, I get the following error (unless I
manually define !tester at the command line first):
> % Not a legal system variable: !TESTER.

Now, I change the program to
> pro tester
> defsysv, "!tester", 222
> help, /system_variables ;this line indirectly tells me about !tester
> end

And it compiles and runs fine, and output suggests that the global
variable has been created: "!TESTER = 222". What the heck is
going on? Is this a bug? Why can't I use a system variable I have
just defined later in the same program? I'm using IDL version 4.0.1.
Maybe I should give up and use a more rational language like C/C++!

-Chad

chad@galileo.ifa.hawaii.edu
http://chipotle.ifa.hawaii.edu/~chad/

Subject: Re: screwy system variables !? [2]
Posted by chase on Thu, 25 Jul 1996 07:00:00 GMT
View Forum Message <> Reply to Message

>>>> > "Chad" == Chad Trujillo <trujillo@xnmusc.mit.edu> writes:
In article <4t71d3$lhe@senator-bedfellow.MIT.EDU> trujillo@xnmusc.mit.edu (Chad Trujillo)
writes:

Chad> (Had some posting problems on the first one)
Chad> It seems like I can't actually create new system variables in a
Chad> function or program and use them later in that same function/program.
Chad> For example, I have the following program (in file tester.pro):

>> pro tester
>> defsysv, "!tester", 222
>> print, !tester   ;this line directly refers to !tester
>> end

Chad> When I try to compile/run it, I get the following error (unless I
Chad> manually define !tester at the command line first):
>> % Not a legal system variable: !TESTER.

The problem is that IDL creates references to variables at compile
time, but the system variable !tester does not exist at compile time.
It will not exist until the defsysv is executed at runtime.

You will either need to perform the defsysv in another procedure that
runs before this one is compiled or, as suggested by someone else, use
an execute() statement wherever you reference the !tester (effectively
execute() delays compilation until run time).

Chris
--
==========
Bldg 24-E188
The Applied Physics Laboratory
The Johns Hopkins University
Laurel, MD 20723-6099
(301)953-6000 x8529
chris.chase@jhuapl.edu

Subject: Re: screwy system variables !?
Posted by thompson on Thu, 25 Jul 1996 07:00:00 GMT
View Forum Message <> Reply to Message

landsman@sorbet.gsfc.nasa.gov (Wayne Landsman (301)-286-3625) writes:

> In article <4t714c$lf2@senator-bedfellow.MIT.EDU>, trujillo@xnmusc.mit.edu (Chad Trujillo) writes...
>> It seems like I can't actually create new system variables in a
>> function or program and use them later in that same function/program.
>> For example, I have the following program (in file tester.pro):
>> pro tester
>> defsysv, "!tester", 222
>> print, !tester   ;this line directly refers to !tester
>> end

>> When I try to compile/run it, I get the following error (unless I
>> manually define !tester at the command line first):
>> % Not a legal system variable: !TESTER.

> You want to have all references to !TESTER in quotes so that the compiler
> won't reject the system variable before the DEFSYSV call is executed. One
> way to do this is as follows:

> pro tester
> status = execute('defsysv, "!tester", 222')
> status = execute('print, !tester')
> end

> Presumably you would want a short program that compiles the necessary
> system variables as the first procedure to be compiled.

> --Wayne Landsman landsman@sorbet.gsfc.nasa.gov

Another way to get around this problem is to put the call to DEFSYSV in your
IDL startup procedure (given by the environment variable IDL_STARTUP). That
way, it'll be guaranteed to be defined before you try to use it, and you won't
need to use calls to EXECUTE.

Bill Thompson

Subject: Re: screwy system variables !? [2]
Posted by peter on Fri, 26 Jul 1996 07:00:00 GMT
View Forum Message <> Reply to Message

Chris Chase SRM (chase@custer.jhuapl.edu) wrote:
: >>>>> "Chad" == Chad Trujillo <trujillo@xnmusc.mit.edu> writes:
: In article <4t71d3$lhe@senator-bedfellow.MIT.EDU> trujillo@xnmusc.mit.edu (Chad Trujillo)
writes:

: Chad> (Had some posting problems on the first one)
: Chad> It seems like I can't actually create new system variables in a
: Chad> function or program and use them later in that same function/program.
: Chad> For example, I have the following program (in file tester.pro):
: >> pro tester
: >> defsysv, "!tester", 222
: >> print, !tester ;this line directly refers to !tester
: >> end

: Chad> When I try to compile/run it, I get the following error (unless I
: Chad> manually define !tester at the command line first):
: >> % Not a legal system variable: !TESTER.

: The problem is that IDL creates references to variables at compile
: time, but the system variable !tester does not exist at compile time.

: It will not exist until the defsysv is executed at runtime.

: You will either need to perform the defsysv in another procedure that
: runs before this one is compiled or, as suggested by someone else, use
: an execute() statement wherever you reference the !tester (effectively
: execute() delays compilation until run time).

Does anybody regularly use system variables this way? Why? Why not a regular variable in a common block?

Peter
