

---

Subject: Re: overriding system functions.

Posted by [Ken Mankoff](#) on Thu, 28 Apr 2005 18:16:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 28 Apr 2005 jlam2223443@yahoo.com wrote:

- > I would love it to have my own preloaded max function that checks
- > if the input is indeed an array and then if it is call the max
- > function or otherwise post an error message and bomb.

You could tell your students to always check for the existence of a "jlam2\_X" function. If they want "max", they should try

IDL> .com jlam2\_max

If there is an error message, use max(), if the compilation works, use jlam2\_max().

You can override any code that is written in IDL. You might be able to do this by appending your library location to the beginning of !PATH. I do it by explicitly compiling my own over-ridden routines (map\_set, write\_gif, write\_mpeg, etc.) in my IDL\_STARTUP file.

-k.

---

---

Subject: Re: overriding system functions.

Posted by [Kenneth P. Bowman](#) on Fri, 29 Apr 2005 01:59:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

In article <1114696363.867468.129920@z14g2000cwz.googlegroups.com>, jlam2223443@yahoo.com wrote:

- > I teach IDL for a small class here and me and my students have trouble
- > with functions such as max and min. Like max expects an array
- > but if I say max(1,2) it runs and on some lists it runs with the
- > wrong results. Fair enough I called it wrong but what I would love
- > it to have my own preloaded max function that checks if the input
- > is indeed an array and then if it is call the max function or otherwise
- > post an error message and bomb. Also, we use -9999 for missing data
- > but sin(-9999) turn it into a reasonable value. I would love my own
- > sin that first check if in the range -20 to 20 radians is the input
- > since if not then probably an error. Any suggestions how I do that?
- >
- > jlam2223443@yahoo.com
- >

If your variables are floating point (not integers), I recommend using NaNs for missing data flags.

i = WHERE(x EQ -9999, count)

```
IF (count GT 0) THEN x[i] = !VALUES.F_NAN
```

This removes the possibility of inadvertently using missing data as valid, and many functions have a NAN keyword (see the manual).

Ken Bowman

---