

---

Subject: defsysv

Posted by [Jeff Hall](#) on Wed, 15 Jan 1997 08:00:00 GMT

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

---

Why does this work on the command line

```
IDL> '!jeff', 38
IDL> print, !jeff
```

when this doesn't work as a pro?

```
pro test
defsysv, '!jeff', 38
print, !jeff
end
```

We are trying to utilize a user-defined system variable to contain a handle value, but there is this problem of being able to define the system variable inside the application. So far, we can only get it to work on the command line, although after the system variable has been defined on the command line then the application works fine. How do you use "defsysv" in a program?

Jeff

---

---

Subject: Re: defsysv

Posted by [Peter Mason](#) on Fri, 17 Jan 1997 08:00:00 GMT

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

---

On Wed, 15 Jan 1997, Jeff Hall wrote:

```
> Why does this work on the command line
>
> IDL> defsysv, '!jeff', 38
> IDL> print, !jeff
>
> when this doesn't work as a pro?
>
> pro test
> defsysv, '!jeff', 38
> print, !jeff
> end
```

I think this comes down to another strange "feature" in the IDL compiler: It seems that IDL recognises system variables at COMPILATION time.

In your routine, !jeff is undefined at compilation time; this is detected and IDL stops with an error message on the line "print, !jeff". (!jeff only gets defined at runtime.)

A sort-of way out is to have a routine which defines system variables but doesn't do anything else with them, and call it before any routine which uses these system variables is compiled.

More simply, define all of your custom system variables at IDL startup (e.g., in a startup file). Then you can reassign their values in your programs (e.g., !jeff=38).

Peter Mason

---