Subject: System Variables: defining and using
Posted by Ray Muzic on Tue, 28 Apr 1998 07:00:00 GMT
View Forum Message <> Reply to Message

I wrote a function is_little_endian to determine if a system uses little endian data representation. (see below) It doesn't compile in IDL4 or IDL5. An error is generated on the return statement with the message that system variable !little_endian does not exist. Apparently IDL requires the system variable to be defined at compile time.

The idea was to, on the first call to this function, determine the endian of the system and store info in a system variable. On subsequent calls the function should note that the
system variable exists and use the stored result.

Is there a way to do this using system variables? I'd prefer to not use common blocks.

I relalize the endian determination is a cheap calculation so there is question as to whether or not saving the result in a system variable is really worthwhile. Nevertheless, I'm interested in the solution in regards to saving results in system variables in other applications.

```
function is_little_endian
; is_little_endian() returns
;       1b (true) if system uses little endian data representation
;       0b (false) otherwise.
;
; Side effects: Creates and sets the value of a sustem variable
!little_endian
; if it does not already exist.
;
;
defsysv,'!little_endian', exists=i
if i ne 1 then defsysv,'!little_endian', (byte(1, 0, 1))(0)
; grab first byte of a 2 byte integer representation of 1 (1st arg. to
byte)

return,!little_endian
```

Ray Muzic
muzic@uhrad.com