Subject: Re: IDL 8.0 compile_opt changes
Posted by monkman on Tue, 05 Jan 2010 21:51:39 GMT
View Forum Message <> Reply to Message

At LASP we have an IDL code base of at least several 100K
lines of code. Much of this code is used in operational
satellite environments, much was written by people whose
main focus isn't programming, and some is very old.
We don't have the resources to look through all of it,
fix what needs to be fixed, test and re-release it all.
Backwards incompatible changes would prevent us from
upgrading to IDL 8.0.

Changing the default integer size from 2 bytes to 4 bytes
would break a lot of our code which deals with a binary
nterface: writing Sybase bcp files, reading CCSDS packets
from a socket, certain bit manipulation code, call_external
etc.

Adding a compile_opt idl1 at the top of every .pro file
wouldn't compile under any IDL version less than than
IDL 8.0, and we run the same code with several versions
of IDL. So this isn't a solution.

Has ITTVIS looked into this possible solution?

What about a new IDL 8.0 command-line option (or a new
built-in system variable) to specify backwards compatibility?
Say if this option were set (or the system variable has
a certain value set by the users IDL startup file), then
the compiler would behave in a backwards compatible way:
parentheses OK for arrays, default ints are 2-bytes.
A given .pro file could then override this by specifying
compile_opt strictarr, and be able to compile new language
features.
If on the other hand, the backwards compatibility option
were *not* set, then the IDL 8.0 compiler would assume
compile_opt idl2 as desired. (But I still don't understand
why changing the default to defint32 is necessary, when the
planned language changes only require a change to strictarr)

If this solution works, the only necessary changes would be
to modify an IDL startup file, or shell scripts which
start IDL. These would provide the new command-line option
or system variable setting to specify backwards compatibility.
This would be much more feasible than changing every file in
the entire code base. And new users of IDL wouldn't have to
set the option, and thus would have the new language features

by default.

-Steve Monk
LASP