Subject: Re: Using IDL from a perl script
Posted by A. Sinan Unur on Fri, 19 Oct 2007 18:37:29 GMT
View Forum Message <> Reply to Message

James Everton <james.everton@gmail.com> wrote in
news:1192816961.304312.109840@q3g2000prf.googlegroups.com:

> I'm developing some web scripts in perl that interact with a database
> through various pre-written IDL procedures here at my work. The
> problem I'm having is getting arguments passed between the two.
> Passing from the perl into the IDL code is easy enough because I'm
> using the open( ) procedure and simply writing the strings as a block:
>
> ```
> open(IDL, "|/usr/local/bin/idl -32");
> print IDL <<"ENDIDL";
> var1 = '$myPerlVar1'
> var2 = '$myPerlVar2'
> .compile my_web_script
> my_web_script, var1, var2
> ENDIDL
> ```

Instead of typing the code yourself, you are having perl type the code.
This is not what I understand from "passing arguments".

> I'm wondering if there is any way to pass from this IDL block (say for
> instance that the my_web_script procedure was actually a function and
> was assigned to a new variable returnVar) out to the containing perl
> script.

There is a serious disconnect in your thinking here. You just opened a
write-only pipe to the /usr/local/bin/idl process. Even if you could
pass some return values (by having the IDL script throw an error), it
would be cumbersome.

Maybe perldoc perlipc can help but you need to improve your
understanding of how interprocess communications work. Why are you
writing the script above? Why wouldn't you just type your IDL program in
an editor rather than generating these scripts using Perl?

Sinan

--
A. Sinan Unur <1usa@llenroc.ude.invalid>
(remove .invalid and reverse each component for email address)
clpmisc guidelines: <URL:http://www.augustmail.com/~tadmc/clpmisc.shtml>