
Subject: Re: Using IDL from a perl script

Posted by [James Everton](#) on Mon, 22 Oct 2007 20:05:05 GMT

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

On Oct 20, 5:16 pm, jkj <ke...@vexona.com> wrote:

> On Oct 19, 1:02 pm, James Everton <james.ever...@gmail.com> wrote:

>

>> Hi everybody,

>

>> 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:

>

> I've done this a bunch and found it to be very useful... here's a bit

> more of what I find works:

>

> use IPC::Open2;

>

> \$progName = "idl";

> open2(READIDL, WRITEIDL, \$progName) or die "Could not begin \"\$progName

> \n";

>

> # Then you can cause IDL to execute commands like this:

> print WRITEIDL "value = 13\n";

>

> # I find that output coming back from IDL is cluttered, so before

> # asking IDL to return a value I ask it to spit out "junk":

> print WRITEIDL "print, \"junk\\n\"";

> while(!(<READIDL> =~ /junk/)){

>

> }

>

> # Now I can get the value I want back asking IDL to output it:

> print WRITEIDL "print, value\n";

> \$value = <READIDL>;

> chomp(\$value);

>

> ...hope that helps,

> -Kevin

Thanks so much! Took me a while to get a feel for how the handles were being set up, but it worked like a charm!

As for the IDL junk, I found that setting my \$progName to 'idl &> /dev/null' got rid of all the junk as well as compilation print outs.

Pipes are your friend :-)

- James
