Subject: Re: Passing arguments at runtime
Posted by Chris Mulliss on Wed, 24 Jul 2002 04:03:38 GMT
View Forum Message <> Reply to Message

Try using environment variables to pass simple parameters into runtime applications.

"Daniel Peduzzi" <peduzzi@attbi.com> wrote in message news:aXV_8.127643$uw.72406@rwcrnsc51.ops.asp.att.net...
> My understanding is that the common (only?) method of passing parameters
> to an IDL program, restored at runtime using the IDL -rt option, is to read them
> from a temporary file (e.g. "input.dat".) I've been using this technique for years,
> and it works well for cases where there is no chance of the same "sav" file being
> restored by multiple processes.
>
> However, dangers arise when multiple processes restore the same "sav"
> file, since the uniquely-named "input.dat" file is in danger of being read
> or changed by the wrong process.
>
> As an example: I have a c-shell script which is kicked off whenever a data file
> appears in a directory. This can happen several times per hour. The script
> takes the data file name, writes it to the temporary "input.dat" file, and starts
> the IDL program which reads the data file name from "input.dat" and immediately
> deletes "input.dat".
>
> To further protect against cases where the IDL program is restored by parallel
> processes, I've tried to name the temporary file something unique, such as
> "input.$$" (where $$ denotes the process ID). Within the IDL program, I get
> the most recently created input.* file via SPAWN, 'ls -1t input.*'. But this can still
> fail when 2 or more processes are started at the same time.
>
> I was wondering how other folks have handled this situation, and if maybe
> there are other solutions which are not file-based and therefore prone to
> synchronization problems.
>
> Dan
>

> ---------------------------------------
> Daniel C. Peduzzi
> peduzzi@attbi.com
> ---------------------------------------
>
>