
Subject: Re: 6.3 reactions?

Posted by [JD Smith](#) on Fri, 05 May 2006 16:24:17 GMT

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

On Fri, 05 May 2006 07:25:57 -0700, codepod wrote:

> JD Smith wrote:

>> On Tue, 02 May 2006 05:52:53 +0000, Michael Galloy wrote:

>

>>> Everyone will have their own favorites, but I think the IDL-IDL bridge
>>> will be a feature that really adds something to IDL. It will allow you
>>> to start new "threads". So, for example, to do some processing while the
>>> controls of your GUI are still responsible to user interaction. Or
>>> simply to farm out several different tasks to run independently.

>>

>> Does this really start an entirely new IDL process, or is there some
>> kind of lightweight user-accessible threading support now available?

>> I'm imagining the kind of havoc which could result if multiple user
>> threads vie for window ids and other common resources.

>>

> For each IDL-IDL Bridge object you create, a child process of your IDL
> session is created and it is executing its own interpreter. IDL
> commands are dispatched in either a synchronous or asynchronous mode
> (which will trigger an event when completed). While a simple interface
> is presented to exchange data between the processes, internally shared
> memory is used so the transfer is fairly efficeint.

>

> As Mike pointed out in his original post, this is a great solution to
> perform background processing while maintaining an interactive user
> interface. Also it provides a quick method to create a "clean" idl
> session to generate save files and the like. And with the proliferation
> of multi-core processors, the bridge provides the IDL user a simple way
> to exploit some of the capabilities of modern hardware.

Very interesting. Since it uses shared memory, can you inherit from
the IDL_IDLBridge class, and have the entire object structure
modifiable from both processes at the same time (which of course would
have its dangers if not orchestrated carefully)? I also take it that
all of the color/window/file handle/etc. resources would be completely
separate, not shared.

JD
