Subject: Re: Help: Communicating between a Windows .exe program and an IDL program

Posted by Robert Barnett on Wed, 19 Jan 2005 21:44:48 GMT

View Forum Message <> Reply to Message

The XML in IDL 6.0 is just a token based parser (XMLSAX). It allows you to set up an event loop of when certain XML elements are parsed, but it doesn't let you explore the bigger picture.

Take the following XML excerpt which might be something you send down the pipe when a certain button is pressed on a certain form.

```
<application_event>
 <form name="mainform">
   <button_click name="submitbutton"/>
 </form>
</application_event>
```

With the XMLSAX you have to extend 'IDLffXMLSAX' and detect when any of these elements have been parsed. This is too difficult and lengthy to show in code.

With the DOM API you might do the following (I can't recall the exact commands)

```
; Somehow XML enters via a LUN device
; Somehow that LUN is read to create the "dom" object
ae = dom -> getElementByName("application_event")
form = ae[0] -> getElementByName("form")
button = form[0] -> getElementByName("button_click")
```

I know that using XMLDOM (or XMLSAX) looks much more complicated than setting up your own way of signalling, but the tools for XML in other languages (especially VB.NET) could make it work your while. I think that VB has some way of automatically generating XML from VB forms and elements. Once again, I don't know the exact details of how this is done, other than I know it can be done.

Using XML also allows you to write a specification for documents using what is called a Document Type Definition (DTD) or style sheet. This is useful because the person programming the TCL/TK could work from the DTDs rather than attempting to emulate the behaviour of your VB program.

There are several ways to get your XML from VB to IDL and back. You can just send it bare over a socket, but most people like to encapsulate it into http requests or as remote method calls (I think it is called SOAP). The problem is that neither are supported by IDL.

XML is quite an involved topic. It's use is often really nice 'in
theory' but it can easily blow out to a very complex problem.

If you do go down this road, I'd really be keen to hear how you go. I've
always been keen to do this with IDL, but I've never had the kind of
project which can't by done using IDL widgets.

Regards,
Robbie

Marc Reinig wrote:

> I know nothing about sockets. I don't disklike Windows. I have been
> developing Windows kernel code and drivers for the past 10 years. I am
> somewhat familiar with VB.NET (novice +). I know nothing about COM+ or
> XML/DOM but can learn.
>
> The only version of IDL I am interested in would be 6.1.1 and beyond.
>
> The reason I thought of sockets is that if I could used it to
communicate
> between IDL code and my app, then someone else could use TCL/TK to do it
> too, or do it on a UNIX platform, which makes the interface widely
usable.
>
> The reason I now have a VB.NET interface, is I can develop faster in
VB.NET
> than IDL and have a wider range of tools to use.
>
> Basicly, what I want to communicate to IDL is button clicks in VB or
send
> file names from VB to IDL and receive information about the completion
> status of a routine from IDL. Data transfer in either direction
would be <
> 100 bytes per second at the worst.
>
> Could you elaborate on XML at little. How would I use that?
>
> -Marco
> ___________________________
> Marc Reinig
> UCO/Lick Observatory
> Laboratory for Adaptive Optics
>
> "Robert Barnett" <retsil@zipworld.com.au> wrote in message
> news:41EB646A.80809@zipworld.com.au...
>

>>Unless you are significantly dislike windows, know nothing about COM+
and
>>nothing about VB then you might want to avoid the use of sockets. VB has
>>features in it to make it accessible from other applications via the
COM+
>>interface. I don't know any details of how this is possible.
>>
>>On the other hand, sockets are probably fine if the problem is
simple. If
>>you're talking about TCP/IP sockets then listening on a socket is not
>>supported (or recommended) in IDL. Thus, you would have to rely on your
>>GUI to take care of listening on a port waiting for IDL to connect.
>>
>>If I were to use sockets I would probably prefer to stream XML (IDL 6.1
>>only) down the socket. I guess that just depends upon your aptitude with
>>XML/DOM as well as future prospects for your project.
>>
>>Asynchronous behaviour can easily be achieved if there is a single
client
>>socket per server socket. Having multiple clients per server can become
>>dangerous if not thought out correctly.
>>
>>I guess it is viable to use sockets. The best solution greatly
depends on
>>the language which the GUI is written in, the API's which are available
>>and whether cross platform compatilibity is desirable.
>>
>>
>>Robbie
>>
>>
>>Marc Reinig wrote:
>>
>>
>>>I have a Windows program that is a GUI.
>>>
>>>I have an IDL program that can do the underlying work. I would like to
>>>control the IDL program with the Windows GUI. My though would be to use
>>>sockets.
>>>
>>>Would this be viable?
>>>
>>>Could I have a widget interface that would work asynchronously to
>>>communicate with the GUI over the socket?
>>>
>>>Any help or recommendations would be appreciated.
>>>
>>>-Marco

>>>
>>>========================
>>>Marc Reinig
>>>UCO/Lick Observatory
>>>Laboratory for Adaptive Optics
>>>
>>>
>>>
>>>
>>>
>>
>>--
>>
>>nrb@ Robbie Barnett
>>imag Research Assistant
>>wsahs Nuclear Medicine & Ultrasound
>>nsw Westmead Hospital
>>gov Sydney Australia
>>au +61 2 9845 7223
>>
>>
>
>

>

--

nrb@
Robbie Barnett
imag
Research Assistant
wsahs
Nuclear Medicine & Ultrasound
nsw
Westmead Hospital
gov
Sydney Australia
au
+61 2 9845 7223