

# Re: Dynamic C to C Data Transfer

---

*Source:* [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2006-12/msg02786.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-12/msg02786.html)

---

- *From:* "Bill Reid" <[hormelfree@xxxxxxxxxxxxxxxxxxx](mailto:hormelfree@xxxxxxxxxxxxxxxxxxx)>
  - *Date:* Mon, 18 Dec 2006 00:34:57 GMT
- 

Tom <[Thomas-911@xxxxxxxxxxxxxxxxxxx](mailto:Thomas-911@xxxxxxxxxxxxxxxxxxx)> wrote in message  
[news:tc2bo29dh62u268q68hak9817ba77c32cc@xxxxxxxxxxx](mailto:news:tc2bo29dh62u268q68hak9817ba77c32cc@xxxxxxxxxxx)

On Sun, 17 Dec 2006 17:36:59 GMT, "Bill Reid"  
<[hormelfree@xxxxxxxxxxxxxxxxxxx](mailto:hormelfree@xxxxxxxxxxxxxxxxxxx)> wrote:

Tom <[Thomas-911@xxxxxxxxxxxxxxxxxxx](mailto:Thomas-911@xxxxxxxxxxxxxxxxxxx)> wrote in message  
[news:6e5ao2lcbal3ofomguqaro5udd98env0he@xxxxxxxxxxx](mailto:news:6e5ao2lcbal3ofomguqaro5udd98env0he@xxxxxxxxxxx)

On Sun, 17 Dec 2006 08:15:46 +0000, Richard Heathfield  
<[rjh@xxxxxxxxxxxxxxxxxxx](mailto:rjh@xxxxxxxxxxxxxxxxxxx)> wrote:

Tom said:

On Sat, 16 Dec 2006  
23:33:27 +0000 (UTC),  
[gazelle@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:gazelle@xxxxxxxxxxxxxxxxxxxxxxxx)  
(Kenny McCormack) wrote:

In article  
<[45845237.1596155@xxxxxxxxxxxxxxxxxxx](mailto:45845237.1596155@xxxxxxxxxxxxxxxxxxx)>,  
Roland  
Pibinger  
<[rpbg123@xxxxxxxxxxx](mailto:rpbg123@xxxxxxxxxxx)>  
wrote:

On  
Sat,  
16  
Dec  
2006  
18:11:49  
GMT,  
Tom  
wrote:

Re: Dynamic C to C Data Transfer

The scenario is: Several networked machines. Each performing stock analysis on individual or a small group of financial instruments

and

then passing buy/sell instructions to an order placing machine.

Is the platform, by any chance, Windows? If so, try using INI files. Works real well.

Thanks for the tip Kenny.

It's a tip that doesn't do you any good, though, always assuming your problem isn't that of opening and writing and reading and closing the files. For that, any old format will work. But

Re: Dynamic C to C Data Transfer

the advice is no more

than

we'd expect from Mr McCormack.

The proper way to do this is via sockets, but yes, you could hack at

it

with

files if you wanted, if you wanted something short-term while you were

busy

learning sockets.

Thanks for your incite Richard.

LOL!!! Thanks for the entertaining spelling!

Of course, all of this sounds like classic job for something like "remote procedure calls" (RPC) just off the top of my head, so I typed "remote procedures" into the Google(TM) search box on my ISP home page and got about 128,000 links in 0.14 seconds.

The top link seems to contain a lot of the "C" code needed to implement client-server RPC, so check it out and see if this is what you want:

<http://www.cs.cf.ac.uk/Dave/C/node33.html>

Or if the ACTUAL problem you are trying to solve can BE solved BEST by using RPC (note carefully the description of RPC in the first paragraph of the link, particularly the part about "network transport independence" in light of the advice here about "sockets"). I'm not completely sure from your description if that is even the case (except you seem to require a "lock" to prevent multiple use of the same file), but if it is, I would think that somewhere in the 128,000 or so links, you'll have enough to proceed, or at least think about...

Now "RPC" is kind of a generalized concept that sort of came out of other concepts such as "distributed computing", and that it was implemented in Unix as "CORBA", and be aware

## Re: Dynamic C to C Data Transfer

that Microsoft has their own version of it (naturally) that has gone through several versions...I've used what they call "DCOM" in the past, so you might want to look at that on the MSDN site (or cite!) for stuff like that...

In any event, back the regularly-scheduled flame war!

For now I am stuck in MicroSoftville. I realize this is disgusting for many of the guru's here.

Actually, it is an advantage, despite what some "gurus" might imply. That's because just about everything you need to write the app is available for FREE and is DOCUMENTED and ACTUALLY WORKS from the MSDN site. Contrast that with all "community effort" crap out there that is "still untested" and "looking for volunteers to document it" (how do you "document" something that doesn't work, or what's the point?).

Substantial portions of the functionality in my current codebase I just downloaded right from the site, seems to work fine...

Also, heterogenous machine architectures make "RPC" somewhat more problematic to implement, which is why "DCOM" is "PC only"...

For now I consider RPC, sockets, file writing hack, and INI techniques to all be possibilities. I'll try to learn enough about each to make the best choice.

The funny thing is, I implemented the EXACT same thing you appear to be working on (a distributed automatic stock trading system) using "DCOM" several years ago...in my case though, I didn't have a choice in the matter, since it was just a demo app as an example of a stock trading API that had added "DCOM" functionality as part of its latest version...

I wonder how many other techniques will surface? In a networking newsgroup it was suggested I consider XML within the .Net framework and using SOAP. <- Another new acronym in the puzzle for me that I have not even googled yet (but I will!)

Heh. As I said, Microsoft has gone through several versions of "RPC", the latest of which was folding "DCOM" into the ".Net framework" and expanding it to include the "standard" XML and SOAP protocols.

## Re: Dynamic C to C Data Transfer

How many different ways are there to do this? Actually, not that many, in the most general sense. "CORBA", "DCOM", and "SOAP" all do pretty much the same thing, which you might suspect because they all include an "O" ("object") in the acronym...but NONE of this is required if you don't REALLY need asynchronous "real-time" execution AND locking capability (which is basically required for "program trading", at least how I implemented the demo app).

I think the XML is way  
out of my league.

Also is overkill, as near as I can tell. You talked about a "40 byte file"), and stock trade orders don't even require that amount of data. You just need a few (maybe just one) remote calls with the necessary arguments (pointer to a 8-character array containing the stock symbol, enum for type of trade, etc.).

That person also seems willing to point me towards  
some useful socket tutorials.

And "sockets" just sounds like massive underkill to me. It's like asking, "how do I drive to Chicago by car?", and somebody gives you instructions for machining an engine block...of course, you didn't help matters any by showing up here with the equivalent of "I have four wheels, how do I make them go roundy-round?"...

---

William Ernest Reid