

Re: What is the story with Server Busy

Source: <http://www.tech-archive.net/Archive/VB/microsoft.public.vb.enterprise/2006-02/msg00003.html>

- *From:* Steve Richfield <Steve@xxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 01 Feb 2006 18:50:39 -0700
-

Michael,

No, just a plain vanilla exe, but with a lot of ActiveX controls in it.

Somewhere something is talking between 2 different processes I believe, which is what is causing your server busy message. If an activeX control calls your app then it will wait for ever without showing a message, but if something cross process calls your app and doesn't get a response then the message will appear. I've noticed this when getting an app to call Excel or Word cross process and Excel or Word is busy.

Interesting, because there are THREE different ActiveX controls for the speech recognition system, which are clearly interacting on the other side from where I am. I thought that this was a bit redundant, but now I can see that this can be worse than just redundant.

However, there are plenty enough simple paging delays to cause failures. COM only allows ~5 seconds to respond, which is simply not enough for large programs.

2. Sometimes it runs into an error because it isn't completely debugged yet. It is important to continue working past errors, as most of them are transitory having to do with speech recognition and synthesis issues. Also, it is important to understand errors when they occur if I am ever going to fix them, so not having things change/crash on an error is really important.

Re: What is the story with Server Busy

When you get these could you popup a non-modal form,

No, because of the mysterious modal form that is hiding somewhere.

add the error to it (so you can see each error with next-previous buttons) and allow your app to continue on?

I'm doing things a little different - in a way that allows me to actually jump in and debug wherever I want! I put up a form for an error, and if I click on a button to debug then I return from the call to the error handler with a result code to execute the following:

```
On Error GoTo 0
```

```
Resume
```

which causes the erroneous line to be re-executed without the error handler enabled, which in the IDE then causes the VB message to be issued. Then, I am stopped on the offending statement in the IDE and I can then debug to my heart's content.

Meanwhile, my real-time routines call an event queuing routine that remembers which events occurred in which order, and with what arguments. Then, when the error has been dealt with, the events are then simulated in their original order, with any additional events being queued until the queues are cleared. Then, the event queue is reset, my error handler returns, and all is back on track.

Note that my error handler is fully re-entrant, which can be pretty important since with all this complexity it is pretty common to encounter another error while dealing with a previous error.

Since this was designed for multi-lingual applications, everything must be translated to the desired target language. This makes for a bunch of additional complexity of its own that is a big source for errors within errors. A couple of days ago my translation DB somehow got corrupted which caused my error handler to die in a recursive descent loop. I normally give up at 5 levels of recursion, but I had missed putting this test into the code that deals with transitory failures, so it just looped until it ran out of stack space.

Again, I don't have any modal windows (that I know of), but the system seems to think that I do! I suspect that the Server Busy message box is a System Modal window that looks modal to my application.

Re: What is the story with Server Busy

That's possible, although I would have thought it would be modal to whatever is calling your app.

I ran into an article (who know how accurate it was) that says that a modal window temporarily sets the Enabled property of all other windows to False. That would probably also stop events from ActiveX controls.

That's true but it would only stop events in the IDE.

Hmmm, of course, things often fail differently in the IDE than they do from an exe, with this being a common cause. Unfortunately, whatever happens in the exe is pretty much of a mystery since I can't do much debugging there.

I know having a messagebox visible will do that but I didn't think a modal form would be a problem.

Aren't they pretty much the same thing?

Yes and no. I think a messagebox will stop cross process calls causing the message you're getting where a modal form won't.

(cringe) Yea, that's my guess. I just hope that it doesn't mutate into something horrible. This highlights probably the BIGGEST problem in linking complex applications together via ActiveX controls, that you reach a level of complexity where you just can't tell what is causing problems. I've been going back and forth between a '98 and an XP machine for development, and using different versions of the speech recognition software, as each has points of sensitivity that the other doesn't. Hopefully this will wring out some of the problems. Even if I am left with some horrible bug, hopefully it will be version dependent, so I can just avoid using a particular version of something.

Re: What is the story with Server Busy

It would be worth checking your API calls to make sure everything is in order. Maybe something that should be defined as long is defined as integer, or a UDT is incorrect.

If only there was something accurate to check against! Like so much other commercial software, the documentation is pretty cryptic and there is no support for any of it. We both know that an argument can be the wrong type and STILL work, yet have nasty side effects later on. Not all "integers" are alike. I watch for any subtle malfunctions suggestive of such things, but there can be cases where everything seemingly works OK.

I've found that passing too-long arguments generally causes less indirect damage than passing too-short arguments, so I always make arguments the longest that is consistent with the documentation, and watch for any problems.

Some ancient history: The original ANSI standard for C was delayed for one year - by me and two other people who were pushing hard to tighten up several things like its screwed up order of operations and arguments types in calls. However, no force of reason could overcome 100,000 prematurely printed copies of the defective draft standard filling a warehouse. Now, ~25 years later, we are STILL being beset by this crap, while Microsoft sends out endless patches for problems that would be impossible with a better language standard. I'd bet that all things considered, that this stupidity cost America >\$1 Billion. I frequently tell them that "I told you so" but STILL they have refused to fix the problems in their standard.

The Ansi C committee literally shredded (as recently confirmed by its president) all of our documents relating to its defective standards, and those same defects remain in the CURRENT standards, not only for C, but for several newer languages that have used C as a (defective) model! If you need a model, then use ADA rather than C/C++/C#.

BTW, I also wrote the ADA linker for ROLM corp.

Steve Richfield

.