

# Re: Embedding Simple MFC GUI app into website

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2006-10/msg01383.html>

---

- *From:* Joseph M. Newcomer <[newcomer@xxxxxxxxxxxxx](mailto:newcomer@xxxxxxxxxxxxx)>
  - *Date:* Fri, 20 Oct 2006 15:16:07 -0400
- 

See below...

On Mon, 9 Oct 2006 14:17:06 -0400, "Pete Delgado" <[Peter.Delgado@xxxxxxxxxx](mailto:Peter.Delgado@xxxxxxxxxx)> wrote:

"Joseph M. Newcomer" <[newcomer@xxxxxxxxxxxxx](mailto:newcomer@xxxxxxxxxxxxx)> wrote in message [news:6rpai2hjbh7gvq49do2jbsmhoqk0j8rrbr@xxxxxxxxxxxx](mailto:news:6rpai2hjbh7gvq49do2jbsmhoqk0j8rrbr@xxxxxxxxxxxx)

On Mon, 2 Oct 2006 12:53:50 -0400, "Pete Delgado" <[Peter.Delgado@xxxxxxxxxx](mailto:Peter.Delgado@xxxxxxxxxx)> wrote:

Comments inline:

"Joseph M. Newcomer" <[newcomer@xxxxxxxxxxxxx](mailto:newcomer@xxxxxxxxxxxxx)> wrote in message [news:d1prh2tap5l5os9ntlod77r1jr774s393t@xxxxxxxxxxxx](mailto:news:d1prh2tap5l5os9ntlod77r1jr774s393t@xxxxxxxxxxxx)

Generally, ActiveX used on Web sites should be referred to by its proper name, "ActiveVirus". When used on Web sites, I consider this a fundamentally evil technology. I have three layers of firewall that strip out all ActiveVirus and JavaVirus from Web sites. Generally, you should assume you have no right to execute anything in any way on the end user's machine. The unfortunate presumption that this should EVER make sense is the source of nearly all invasive malware.

## Re: Embedding Simple MFC GUI app into website

Joe, I understand that you dislike the technologies you've listed above and that you have some valid points for your opinion, but to say that a particular technology is "evil" goes beyond common sense and increases the likelihood that your warning will be ignored as hyperbole.

In addition, if your firewall is stripping out ActiveX controls, then how are you using Windows Update? Or perhaps you are back on UNIX? ;)

\*\*\*\*\*

I don't use Windows Update. I will NOT use any technology that requires ActiveVirus.

The term "evil" is carefully calculated. A technology whose main purpose is to run unmanaged code without restriction on a target machine is inherently evil.

Are you saying that any technology that allows you to download a native executable is evil? FTP is thus considered evil because I can download Visual Studio Express from Microsoft and after the code is downloaded and run, it can do anything that it wants to my system?

\*\*\*\*

Note the qualifications I gave. The technology "whose main purpose is to run unmanaged code without restriction on a target machine", with respect to browser and email technologies such as ActiveVirus and JavaVirus.

The role of FTP is to transfer bits, not to cause the downloaded code to run without my permission or control. And I \*can\* sandbox such code because, if I don't trust it, I can run it in a restricted account. That is \*my\* choice. The client-side scripting technologies run code without my permission, without my control, and with no ability to audit their consequences. So they don't run without my permission, and I \*never\* give my permission.

\*\*\*\*

At what point does the user take some responsibility for what he/she has downloaded or allowed to exist on the computer?

\*\*\*\*

The point is that I want that responsibility, and client-side scripting technologies, as currently implemented, deny me the ability to take responsibility.

\*\*\*\*

Re: Embedding Simple MFC GUI app into website

It shows a lack of any kind of responsibility. A "fix" was kludged up to use code signing, but there are some fascinating means of creating signed exploits.

The real problem is that a technology that can download, without my permission, and execute, without my control, any code from the Internet, can only be described as "evil".

Without your permission?

\*\*\*\*

Without my permission. That is the \*default\* for all these technologies. I have no interest in having a lot of dialog boxes pop up (six–eight times for some Web pages) only to have to deny each of them individually, so I just do a blanket deny. That's my view of permission: NO WEB SITE EVER HAS MY PERMISSION TO RUN CLIENT–SIDE SCRIPTING.

The people who think client–side scripting is mandatory show such a profound lack of consideration for the end user that the most polite term that can be applied to these people is "sociopathic".

\*\*\*\*\*

I suggest that you try to download an ActiveX control from the Microsoft web site and see what happens. Unless you have modified your configuration for IE I believe that you will be notified that the web page wishes to load the control.

\*\*\*\*\*

I can select that option. So why should I grant them permission? Remember these are the same people who, in at least two major conferences, distributed CDs containing viruses, and made no attempt to replace them.

\*\*\*\*

Even the Windows Update site and the "Genuine Windows Advantage" control will trigger IE to prompt you to accept or deny the control.

\*\*\*\*

Not the "out of the box" configuration.

\*\*\*\*

My big problem with such technologies is that the users become so accustomed to clicking "OK" and "Accept" on the numerous security dialogs and prompts that Windows throws at them, that they inadvertently accept controls and downloads that they really shouldn't because they just want the dialogs to go away.

## Re: Embedding Simple MFC GUI app into website

\*\*\*\*

Which is why the control has to be automated completely. The problem with the current technology is that I have to give pre-approval to a control whose behavior I do not know and cannot determine. If I simply could set defaults so that a control was not permitted to read or write files on my system, was not permitted to modify my Registry, etc., and ran under a specific set of privileges, where I could set the controls and enable, disable, or demand prompts for each operation (e.g., "This script wishes to [create,modify, read] the file <full path here>" or "This script wishes to modify your Registry <full path here>, old value, new value" this would be a good start.

\*\*\*\*\*

The key here is that there is absolutely no way for me to force such code to be sandboxed, at least not without investing in relatively expensive third-party products.

There are many products out there such as Virtual PC that can be used for such purposes. Virtual PC is now free as are some of the other products.

\*\*\*\*

Note that the way virtual PCs work is that they roll back all state. So if I discover damage next week, I lose a week's work? I don't think so.

\*\*\*\*

JavaScript and ActiveX are both technologies whose execution cannot be restricted and sandboxed, and in fact, are not inherently restricted or sandboxed.

That is a grossly inaccurate statement. The execution of both can be restricted and by default are. Take a look at the help files for IE and the various KB articles such as KB240797 for more information on how to "lock down" IE.

\*\*\*\*

I do not program in assembly code. Note that this article (which I had seen before) requires that I individually disable the ActiveX controls one-at-a-time. This is not practical. I have said for years the problems with OS security is that we are doing security by thinking like assembly-code programmers (kill bits, ACLs, etc.) instead of having a 4GL interface for doing this. The fact that this gives specific advice about how dangerous it is to edit the Registry manually tells right away that this is at best a kludge.

I don't want to kill them; I want to limit what they do. So this mechanism is rather

Re: Embedding Simple MFC GUI app into website

hopeless anyway. It doesn't solve most problems, and it doesn't solve the global problem, only the problem for one control at a time. Sort of like programming in hexadecimal instead of using an assembler.

\*\*\*\*

most of the exploits that require security patches turn out to be triggered by such exploits.

There was a wave of such exploits some time ago, but it seems to me that the majority of the exploits uncovered over the past year or so are unrelated to either JavaScript or ActiveX. I admit though, I am too lazy to take a look to verify my suspicion!

\*\*\*\*

I looked a few months ago on the Norton Antivirus Web site, and these exploits were still documented.

\*\*\*\*

\*\*\*\*\*

Do everything server-side. That's safe.

I would say that it's \*safer\*. The only way to be "safe" is to not use the web at all! Recall the bug in the jpeg software that Microsoft patched last year?

\*\*\*\*

Another clear example of a fundamentally incompetent design; the jpeg file format allowed for arbitrary code to be included. Only a committee consisting entirely of idiots would have allowed this design to escape. This kind of design illustrates the complete lack of responsibility going on right now in the world.

## Re: Embedding Simple MFC GUI app into website

Joe,

Not everyone who creates a bug or writes a piece of bad code is an \*idiot\*.

The JPEG code was written at a time when functionality and speed were of far greater concern than security. To hold the developers to the security standards of today is unfair and dishonest.

\*\*\*\*

No, the JPEG code was written by people who had ABSOLUTELY NO CONCEPT OF SECURITY. This was an stupid decision, deeply irresponsible, and no one should have allowed such a feature to have been implemented.

We knew how to embed code in images in 1976 or thereabouts, and anyone who had a shred of common sense would have known how dangerous this was. Several people at CMU were nuked by image malware virus before the feature was disabled by having filters on incoming Internet connections to remove screen control sequences from finger pages, the predecessors of Web pages. (Note that the formal RFCs for Finger came out in 1977, but the protocol dates to 1971—I just checked Wikipedia. I can date it to 1976 because I can date it to a particular location of my office in 1975–1976 and the acquisition of a particular terminal type at CMU. No, I wasn't the victim, but several friends were). So the possibilities were well-known, well-understood, and acknowledged as dangerous in the 1970s.

Perhaps "idiot" is too strong. How about "irresponsible children with no adult supervision"?

I'm not talking about the security standards of 2006. I'm talking about the security standards of 1977. Since it would make no sense to put executable code in during that era, putting it into a JPEG file (the format first came out in the mid-1980s) makes no sense, particularly because in that era there were so many \*different\* machines! So when was it added? By the late 1980s, code exploits were know (the Macintosh was particularly vulnerable, and code in a JPEG was no different from code in a Mac resource! So the problem was well-understood by 1988, when I bought my first Mac and it came with an antivirus program!) By 1990, the folks in the Andrew project at the Information Technology Center at CMU had been embedding executable code in email as a regular thing, but they understood that it required security validation and it was executed sandboxed. They understood the problem in the late 1980s, and had published papers on it. Their security decisions were made around the same time as JPEG was first invented, so the problem was well-understood BEFORE THAT TIME!

Interestingly, one of the CMU researchers was told by someone at Microsoft that they liked the Andrew ideas. So they took the base technology ideas but got rid of all that security stuff because it just got in the way. Perhaps "idiot" \*was\* the right term after all...

\*\*\*\*\*

Today's corporate developers often are driven by schedules and feature sets. While I believe that the majority of the professionals working today are fully capable of creating secure applications, the fact is that unless your company has the luxury of unlimited time and budget, at some point compromises have to be made. Even the best and most secure design can be thwarted by a rushed implementation.

Re: Embedding Simple MFC GUI app into website

\*\*\*\*

There is no excuse for a lot of the egregious security holes that have been created. ActiveVirus was "rushed", created by unsupervised children and pushed by incompetent marketers, neither of which was qualified to understand security. As we left the PDC where it was introduced, I turned to my friend Ed and he said "The future of virus delivery" and I said "the future of industrial espionage". We were both right. If we could see this five minutes into the presentation, where was the adult supervision at Microsoft that should have killed this technology before it ever saw the light of day?

\*\*\*\*

\*\*\*\*

Note that possession of a certificate is only barely acceptable (who verifies that the certificate was issued to a valid address and not, say, to a company located on a vacant lot in the Cayman Islands?).

Which is exactly why I stated that the certificate must come from a "well-known authority". Companies such as Verisign make their living out of being trusted authorities. If they don't engender trust and confidence, they go under.

\*\*\*\*

And what, exactly, does a certificate prove? It proves that Verisign believes the company to be legitimate. If my goal is industrial espionage, I could get a certificate. By the time the espionage is discovered and traced to me, I'm living in some country without extradition treaties. For that matter, suppose my legitimate ActiveX control contains code that I didn't put there? The number of workarounds to certificates is quite large, and I know how to do several of them.

\*\*\*\*

A certificate proves that the trusted authority believes that you are who

Re: Embedding Simple MFC GUI app into website

you say you are \*and\* that the code has not been modified since it was signed.

\*\*\*\*

And why does this tell someone else they should trust me? Why is it that saying who I am gives anyone a sense of "trust"?

\*\*\*\*

We could play the "what if" game all night Joe, but in the end there will always be a certain amount of risk. The question is do you completely go dark or do you minimize your exposure?

\*\*\*\*

I find the level of risk to be unacceptable. I consider the attitude that I should accept such high risks to be sick. I consider the attitude that DEMANDS that I accept these risks to be sociopathic.

Here's the deal: FedEx will no longer deliver packages to your home unless you leave the door unlocked. Not to worry, all their drivers are bonded, and would never take anything. And if you are worried about someone else coming in, hey, feel free to hire a security guard to watch your door all day. How long would FedEx stay in business after such a decision? Yet computer users accept similar risks every day. What's Wrong With This Picture?

\*\*\*\*

\*\*\*\*

The first thing you should assume when the idea crosses your mind to use ActiveVirus is "I am making a fundamental error here" and proceed with that assumption as the basis of all decisions.

Which is why I tried to gently lead the OP in a different direction. The security problems with ActiveX along with the fact that nobody will want to have to install it should leave the OP with the impression that there is likely a better solution.

Re: Embedding Simple MFC GUI app into website

\*\*\*\*

There is always a better solution.

\*\*\*\*

Perhaps, but is there always a better solution that fits into the timeframe and budget of your company? Is there a better, more elegant more scalable solution or is there a duct-tape solution that eliminates one form of risk only to introduce 10 others?

\*\*\*\*

Server-side scripting removes the risk. I have several friends who make their livings doing server-side scripting, and they uniformly believe this is the better method for security. It is also cheaper to implement (one company had been doing client-side scripting and gave it up because of the high costs of development and support, and they moved to server-side scripting. They could do more, in less time, at lower cost, with higher reliability and a better user experience).

\*\*\*\*

I believe that ActiveX still has a place but that in \*most\* cases that it is proposed there is indeed a better, safer solution that provides the required functionality.

-Pete

Joseph M. Newcomer [MVP]  
email: newcomer@xxxxxxxxxxxxx  
Web: <http://www.flounder.com>  
MVP Tips: [http://www.flounder.com/mvp\\_tips.htm](http://www.flounder.com/mvp_tips.htm)

.