

From: Brad Silverberg  
Sent: Friday, November 01, 1996 10:45 AM  
To: Steven Sinofsky  
Subject: RE: Java thoughts

what part don't you agree with? you don't think people will be able to write real code with trics? if not why not? think of trident as the new user+gdi. the scripting and components can do anything you want including "real code".

there is an incredibly strong desire by developers and lops for cross platform. there isn't a day that goes by where someone important is switching over to java.

From: Steven Sinofsky  
Sent: Thursday, October 31, 1996 4:15 PM  
To: Brad Silverberg  
Subject: RE: Java thoughts

I don't agree. I think that you are underestimating the desire and need of people to write real code. I used to have endless meetings with programmers trying to get them to use VB because it was so much better at doing what they wanted, so much faster, etc. but in the end they seemed to always go with the "real" tool.

I think any successful strategy will do both well and for the long term. There is no one tool to solve all problems.

-----Original Message-----  
From: Brad Silverberg  
Sent: Thursday, October 31, 1996 3:04 PM  
To: Steven Sinofsky  
Subject: RE: Java thoughts

there are a lot of tough issues. we have not yet settled on an api strategy. there is some lively debate going on, and in fact bens came to me this am with the proposal that we do do mfc for java. I'll fwd. I do believe that the ultimate api strategy will be one based on trics (trident+scripting+components) but shorter term mfc/java may be a more effective tactic.

this is one reason, to be honest, steve, I am excited about the reorg. I know you probably don't want to hear this but some of the challenges we face as a company are incredibly difficult; I think you'd have a lot to contribute directly to these api discussions. I am excited about some cross fertilization and getting more smart minds on our tough problems both apps and platforms. I have seen with ie3 how much smart people can do when they are unleashed from the organizational and strategic constipation that has paralyzed so much of this company. we have been way way way too inwardly focused and not enough on the outside world and competition.

From: Steven Sinofsky  
Sent: Thursday, October 31, 1996 2:08 PM  
To: Brad Silverberg  
Subject: FW: Java thoughts

arghl

-----Original Message-----  
From: Steven Sinofsky  
Sent: Saturday, February 24, 1996 6:10 PM  
To: Denis Gilbert (Kenix)  
Subject: Java thoughts

MSS 0118220  
CONFIDENTIAL

I've been thinking (oh no) a bunch lately about this Java thing and what we're doing (as far as I can tell). Excuse me for being random...

I'm really caught by how I think Java is really just the once a decade "programmer full employment act" where all the programmers get together and change languages just so that they can all feel like they are

Exhibit No. 20  
Wit. Silverberg  
Dorothy Stiles, CRR  
7/8/96

MS98 0160269  
CONFIDENTIAL

TXAG2 0001855  
CONFIDENTIAL

making progress. Seeing Bruce Eckel giving a talk on "Java for C++ Programmers" reminded me of a talk he gave in 1990 called "C++ for C Programmers". I bet if Bruce was writing books in 1980 he'd be writing "C for Pascal programmers".

The more things change the more they stay the same.

Dean explained to me that we're not doing an MFC-like class library for Java but writing a whole new library around Forms<sup>3</sup> and OLE (sure it will leverage MFC experience, but it isn't a total clone). I guess this seems pretty strange to me (maybe I misunderstood and the whole thing is moot). We created MFC to help with two things:

- (1) Make it easy to write Windows programs by leveraging the general understanding of the C SDK and Windows in general (Petzold in particular)
- (2) Make it easy to write C++ (OO) programs by leveraging your C knowledge

I think moving to Java requires the same sort of thinking. For me the real winner in the Java tools world will be the company that

- (a) has a Java development environment that is great (everyone is trying to do that and I just assume we'll be more than competitive in that arena) and leverages folks understanding of writing C/C++ programs. In particular the vendor that provides the best tools for calling C/C++/MFC APIs from within Java and vice versa will win (just like we made 100% sure you could call Win32 from MFC).
- (b) has a Java library that is the easiest to write programs that are cool and competitive with regular Windows apps

When I look at this it just screams for an approach that puts MFC into Java. Not something "like" MFC, but MFC itself. Imagine if there was a set of Java classes that are pretty much the same as MFC classes. This would mean that the average MFC programmer could start cranking out Java apps in no time flat and all the people that teach MFC could be teaching MFC-Java tomorrow without a big learning curve. Basically this is what we tried to do with MFC-make it so the SDK (or C++) people of the world didn't have to learn too much new stuff in order to make the big leap to our world. All along the way we bowed to the object bigots and to the portability police with talk of VLM and licensing (which finally came to be!). In the end, MFC became the only way to use C++ for Windows.

We shouldn't get caught up in the excitement of using the class library to force each and every new technology on people-there will be another 18-24 months where people are just trying to get their head wrapped around new syntax, runtimes, and paradigms. So if we are first out there with an MFC library then there will be that many more people who can use our tools and will be Java programmers like we want and we can then throw stuff like OLE/COM/Forms<sup>3</sup> at them.

I feel that if we try to accomplish two things at once (the transition to Java and the transition to all the new infrastructure) we're likely to overload people and they will fall back on the Sun AWT way of doing things and we will have lost the API war. People need simple concepts that leverage what they know. Right now AWT is random and doesn't do enough to write apps with-MFC is well understood and does what people need.

I also think that the one thing that should be done differently with Java is build in data binding earlier than we ended up doing in MFC. Java is just going to replace VB over time for a lot of apps-it really seems to me that will be natural. VB as a language has failed to evolve and really is archaic and there is no reason to continue to suffer so much (for example, classes really are a good thing). I think this could really be a final nail in the VB coffin.

So I think if we just had MFC2 with data binding as a set of Java classes in 6 months we'd be in really great shape.

In the end this is all about making sure we have strategic APIs. Strategic APIs are those that are clear, easy to explain, and easy for people to use. OLE does not do that, so that is why no OLE interfaces are really strategic. MFC has a better way of expressing those things and it is strategic. But in order for something to really be strategic it has to map very well to the OS and I think it is premature to think that all these new technologies are what people want right away.

I think over time (maybe in MFC Java 2.0) you can add a new CForms3View class and support for that, but the immediate need is to avoid losing all the MFC people to AWT.

Just some thoughts...

MSS 0118221  
CONFIDENTIAL

TXAG2 0001856  
CONFIDENTIAL

MS98 0160270  
CONFIDENTIAL