



From: chrisjo
Sent: Monday, September 04, 1995 3:15 PM
To: victors; johnlu; brads; bens; thomasre; darrylr; paulma
Cc: chrisjo
Subject: RE: Anti-Java strategy memo

Importance: High

Some additional thoughts... these all are based on my conclusion that Java is already here, and we need to move down the embrace/extend path.

1 Our browser must support Java. Java is already here. Netscape will add Java support in their Win95 version before the end of the year, and sites (like StarWave) will begin posting Java content shortly after. I believe there will be a proliferation of these objects regardless of what kind of competitor we introduce, and therefore Java will become a de facto standard we have to support. We must move quickly here to decide exactly how to do this, in terms of technology, distribution, and licensing.

First question is how to do this architecturally. One approach we could pursue is to host OCX's in our browser, and then incorporate Java objects through an OCX wrapper. Starwave has indicated they are writing that wrapper, so we may be able to use that code as the wrapper. This doesn't solve the problem of class libraries/HotJava, nor does it address the object model or default support that Netscape may be providing. We need more information about how they are doing this ASAP. Another option here is to allow a plug-in Java DocObj viewer, which we have someone else (Sun? Starwave?) write. This would let us view Java documents without having to support Java fully ourselves, and also put someone else in control of these "apps". Second, and equally pressing, question is one of licensing and distribution. Do we bundle their runtime in our browser? Do we license code from Sun or write our own? Do we point users to a web site to download the code themselves? There are a bunch of options, but the answer really revolves around our company strategy. Given that we don't think Java is going to be around, or that we don't want to encourage it, we should figure out a way to have our browser sniff the "applet" tag in HTML (Java object), then prompt the user to fetch the runtime from Sun.

2 We need to lead in areas where Java/Sun is weak. I think this is key -- we will not be able to produce a Java competitor, but we should flank Java in areas it will not be able to quickly move into. I think these are

Signed x86 code. There's a lot of this out there, and our tools generate it. We should have some way that ISV's can register/sign code as "secure" and we will "guarantee" it. People will want to download native code, and one group at MS should lead the way signing both internal and external code for ISV's. We may even want to provide this "for free" to people who write OCX controls or other code that plugs into our browser and tools. The MSN group is thinking about this today, but this must become top of mind and we have to staff the effort.

Simple scripting. We should introduce the web scripting language. This should be VB in syntax, although implementation may be

different. This script should be downloaded in text form, so that it is easy for people on the web to copy and paste it, and only do a simple set of things. We could make controls talk to each other, and in most cases the things you want to do are simple (on click set this property to that property, etc). This language could be used in our browser/BlackBird to script OCX's that we have signed, or Java OCX's.

One strategy is to jump on the Java bandwagon and try to take control of the class libraries and runtime. This may be the best alternative, but this decision impacts across the company.

ATTORNEYS ONLY

MSS 5040320
CONFIDENTIAL

MS98 0164419
CONFIDENTIAL

3. We should consider support for Java as a platform. As a company, we have two options for embracing and extending Java: (1) we take control of it and add Windows specific classes, or (2) we "sandbox" it, slow it down, and restrict it to a particular domain, betting that we can bring our technology to bear quickly enough to minimize the impact. While I would like to pick (2), my personal feeling is that we should strongly consider (1) - namely fully supporting Java and extending it in a Windows/Microsoft way.

Why? First, I think that every single one of our competitors is going to jump on Java as the way to beat Microsoft. There are no fundamental flaws in the runtime, language, or classes which prevent this, and you'll see companies like Apple, Borland, Sun, and Netscape each working in their own way to extend Java.

Second, they will have great distribution in the client, so that there is going to be a huge installed base (even if only on Win95) of clients that can run this code. In addition, it lets publishers write once for Win/Mac, and multimedia is one area where people have been writing x platform for quite some time (director).

Third, the Web standards bodies will probably jump on this, as well as scientific research efforts. NCSA is already pulling their top people into a Java team focused on communications/collaboration. Given proper license, the standards bodies may promote Java as "the" cross platform VM.

It is possible that wrapping Java in OCX's combined with great execution from all product groups will be a good enough alternative, and we can sandbox them. We can also wait and see on this. However, it is probably worth our time/resources to have some people (DarrinM) think about what we would do technically if we wanted to take control. Would we change VB x-codes to run on the Java VM? Would we add COM to Java so it could call other objects? We could have a killer story here if we went this route, but this is risky as well given that we might only entrench/validate Java without gaining control.

Chris

-----  
From: Darry Rubin  
Sent: Tuesday, August 22, 1995 4:46 PM  
To: brads; johnlu; paulma

Cc: ben : thomasre  
Subject: Anti-Java strategy memo

4, 3, 1

ATTORNEYS ONLY

MSS 5040321  
CONFIDENTIAL

MS98 0164420  
CONFIDENTIAL