VI. Microsoft Used Predatory and Anticompetitive Conduct to Impede Other Platform Threats as Well, Thereby Further Entrenching Its Operating System Monopoly

A. Microsoft responded to the threat that Java posed to the applications barrier to entry by engaging in predatory and anticompetitive conduct

318. As explained, Java technology, both alone and in concert with non-Microsoft browsers, poses a threat to Microsoft's operating system monopoly because it holds out the possibility of an effective cross-platform middleware that can significantly reduce the applications barrier to entry.

i. See supra Part III.C.; ¶¶ 57-59.

319. Microsoft recognized the Java threat and, in conjunction with its effort to blunt the browser threat, engaged in a series of actions designed to interfere with the development, distribution, and usage of cross-platform Java.

319.1. Microsoft developed and then widely distributed, in part through Windows, a "polluted" version of Java that is not cross-platform.

i. See infra Part VI A.1.; ¶¶ 320-321.

319.2. Microsoft's purpose in creating and widely disseminating its version of Java was to fragment cross-platform Java and thus hinder the threat Java could pose to the applications barriers to entry.

i. See infra Part VI.A.2.; ¶¶ 324-326.

319.3. In addition to polluting Java, Microsoft engaged in anticompetitive conduct designed to cripple cross-platform Java.

i. See infra Part VI.A.3.b.(1); ¶¶ 330-332.

319.3.1. Microsoft, through its predatory campaign against Netscape, weakened cross-platform Java's principal distribution vehicle.

319.3.2. Microsoft used its monopoly power to force third parties to support its version of Java and to mislead developers into creating Windows-specific programs.

319.3.3. Microsoft induced third parties not to support cross-platform Java and to support exclusively Microsoft's Java implementation.

1. Microsoft "polluted" Java by developing and distributing a version that is not cross-platform

320. The first step Microsoft took to diminish the cross-platform threat Java posed to the applications barrier to entry was, in its own words, to "pollute" Java by developing a variation of Java that is not cross-platform.

i. An internal planning memorandum for Microsoft's Java development tools for 1998 reaffirmed the objective: "Kill cross-platform Java by grow the polluted Java market." GX 259, at 1.

ii. James Gosling testified: "Microsoft has made an incompatible implementation of the Java technology that is not cross-platform, but instead is dependent on the Windows operating system platform and Microsoft's proprietary technology . . . . Microsoft has designed its implementation of the Java technology to impair the ability of programs written to that implementation to run on non-Microsoft platforms, or even to operate properly on the JVMs sold by other vendors for PCs running Windows." Gosling Dir. ¶¶ 54-55. As Gosling further explained, this effort was "part of the Microsoft program of making it so that people could port into the Microsoft world but then sort of swallow the pill and not be able to get out." Gosling, 12/10/98am, at 37:4-7.

iii. An internal Boeing Corporation presentation summarized the overall impact of Microsoft's "pollution" of Java (implemented through the inclusion of Microsoft's Java virtual machine in Internet Explorer) in December 1997: "Internet Explorer will allow the development of Java applications that are <u>dependent on the Windows platform</u>. These applications may not run on other platforms." GX 639, at TBC 000569 (emphasis in original).

320.1. <u>First</u>, Microsoft refused to support standard Java native code interfaces and developed, instead, Windows-specific interfaces.

320.1.1. As part of the evolution of Java, Sun and several of its Java licensees developed a standard Java API called the "Java Native Interface" (JNI). JNI enables Java programs to draw upon code unique to particular operating systems in a way designed to minimize the cost of porting the programs to the various operating systems.

i. "The Java technology is in the process of maturing, and occasionally a developer will encounter a situation in which the programmer needs to write a portion of a program in a different programming language in order to access functionality not yet supported in the Java technology, but which may be available in the underlying (‘native’) operating system or hardware. To do this, the Java technology includes a ‘Java Native Interface’ (‘JNI’). JNI is a standard Java API that acts as a link between the JVM and the platform-specific code included in an application to perform the particular operating-system function. JNI thus gives programmers a way to use native platform functionality with their Java-based software." Gosling Dir. ¶ 15; Gosling, 12/9/98pm, at 50:23 - 51:10 ("JNI is one that we, along with IBM and Netscape and a number of other companies, worked to make JNI work with multiple Java VMs."); Gosling, 12/9/98pm, at 44:16 - 45:21 (explaining that, “while JNI doesn't solve the problem of making native code transparent” it “goes a long way towards making that job much less onerous, much less costly for developers, so that they can take their . . . native code and move it from one system to the next much more cost-effectively.”).

ii. Sun’s public statements emphasize the cross-platform significance of the JNI approach to native interfaces. DX 2014 (Sun web site, "100% Pure Java Cookbook:" "The Java Native Method Interface (JNI) is not a way to make native code platform-independent; it is a way to make it easy to port native code."); DX 1944 (Sun press release, April 97: "The JNI standard may give a native library its 'best chance' to run in a given JVM, according to JavaSoft.").

iii. See also Gosling, 12/3/98am, at 9:1-5

Soyring, 11/18/98am, at 73:19 - 76:6 (explaining that JNI is a "tremendous value" because it offers "the possibility of writing an application once, compiling it once and using the jni interface to access functions on different operating systems"); Fisher, 1/7/99pm, at 27:14-20 (testifying that JNI permits developers to call native OS functions "in a way that the code written to the JNI is, as it were, hived off, so that if you want to write for a Java virtual machine for a different underlying platform, it is relatively easy to port it").

320.1.1A. Microsoft's assertion that a standard native code interface like JNI "has nothing to do with writing cross-platform Java applications" (MPF ¶ 852) is wrong because, as explained above, one of the primary benefits of a standard native code interface like JNI is to reduce the difficulty and costs of porting Java applications that rely on native code to other platforms.

i. <u>**See**</u> <u>**supra**</u> Part VI.A.1; ¶ 320.1.1ii.

ii. **Gosling emphasized the importance of developing native interfaces in a way that maximizes the portability of Java programs. <u>See</u> Gosling, 12/9/98pm, at 44:16 - 45:21.**

iii. **Soyring explained how JNI, though designed to enable programmers to reuse platform-specific code, facilitates cross-platform portability: "Whenever we've used it -- and we've used the JNI function on projects that I have managed with our clients -- we need to make sure that the function is implemented consistently across the various operating systems on which they want to run the Java application. And we've been able to successfully demonstrate with a variety of our clients around the world that we can write and compile one Java application which uses the JNI programming interface on multiple operating systems." Soyring, 11/18/98am, at 73:23 - 74:6.**

320.1.2. Microsoft, however, refused to support JNI in its implementation of Java (until ordered to do so by the District Court for the Northern District of California).

i. "Microsoft omitted from its implementation of the Java technology a standard API called 'JNI,' that permits platform-specific software code to interact with Java code in a program." Gosling Dir. ¶ 58.

ii. Sun Microsystems, Inc. v. Microsoft Corp., 21 F. Supp. 2d 1109, 1127-28 (N.D. Cal. 1998) (requiring Microsoft to include JNI in its Java environment).

320.1.3. Instead, Microsoft developed its own native interfaces, known as "J/Direct," the "Raw Native Interface" (RNI), and "Java/COM." These interfaces are designed to allow developers to link directly to Windows-specific DLLs (in the case of J/Direct), Microsoft's proprietary and Windows-specific Component Object Model (COM) architecture (in the case of Java/COM), and other Windows-specific code (in the case of RNI).

i. Gosling testified: "Microsoft substituted its own proprietary interfaces, called RNI, J/Direct, and @COM. By omitting the standard API, Microsoft prevents developers from writing a Java-based program that can run on every JVM implementation. If the developer uses JNI, the software will not run on Microsoft's JVM; if the developer uses Microsoft's proprietary RNI, J/Direct, or @COM interfaces, the software will not run on any other JVM. Moreover, Microsoft's proprietary interfaces require the use of Microsoft's development tools." Gosling Dir. ¶ 58.

ii. Robert Muglia admitted that J/Direct (and its accompanying compiler directive, @DLL) is designed "to provide developers with additional development tools that make it easier, faster and more efficient for them to use the Java programming language to build full-featured Windows applications." Muglia Dir. ¶ 122 (emphasis added).

iii. As Intel's Alan Holzman noted at a meeting with Microsoft in April 1996: "MS has changed native interfaces. Not compliant with Sun's current native interface at the binary level: As part of the rewriting the Java VM, MS has completely changed the internal object model to accomodate COM. We think they have not told Sun and this may be an issue for Sun as well as us if Intel and MS give a single optimized Intel Architecture Java RT back to Sun. The current method for calling native methods in the MS Java VM is different and not compatible with Sun's. MS will provide include files that allow you to bridge between the two at the source code level but not at the binary level." GX 566.

iv. See also Muglia, 2/26/99pm, at 81:21 - 82:16 (explaining J/Direct and testifying: "I'm not aware of any other operating systems that support J/Direct, although I do know that we openly publish the specifications so that others could take advantage of it."); Gosling, 12/10/98am, at 27:17 - 28:23 (making Microsoft's native interfaces available to work with other operating systems is "not a trivial engineering effort" inasmuch as "there is essentially no way to implement them without reimplementing some significant part of Windows on whatever your system happens to be"); Gosling, 12/10/98am, at 43:20 - 44:4 (testifying that "we certainly did have discussions about building a solution based on COM." Gosling further testifying that: "but as I recall, there was no real interoperability that was ultimately possible, since COM only existed on the Windows platform."); Gosling, 12/9/98pm, at 55:24 - 56:14 (MS's RNI interface was based on COM, and "didn't satisfy our needs for portability because it was an interface that could only work on Microsoft's operating system since that was the only place where . . . COM support existed."); Gosling, 12/9/98pm, at 76:12-15 (Microsoft talks about making COM cross-platform, but hasn't done so).

320.2. Second, Microsoft created Windows-specific extensions to Java in its Java development tool products which, when utilized by developers, caused programs to work with only the Windows operating system.

i. Gosling testified: "Microsoft extended the Java programming language in ways supported only by Microsoft's incompatible implementation of the Java technology. This is analogous to adding to the English language words and phrases that cannot be understood by anyone else. Specifically, Microsoft did two things. Microsoft added support for incompatible 'keywords,' or additional Java programming instructions, to its implementation. And Microsoft added to its implementation support for 'compiler directives,' or special comments inserted in Java sourcecode that alter the behavior of the code when it is compiled into Java bytecode. Among other things, Microsoft's compiler directives permit the Java Virtual Machine to interact with or 'call' proprietary Microsoft Windows APIs. Only Microsoft's incompatible implementation recognizes these idiosyncratic keywords and compiler directives. Among other things, some of these extensions prevent programs from being compiled by non-Microsoft compilers, and all such extensions prevent programs from running as intended on non-Microsoft JVMs." Gosling Dir. ¶ 58.

ii. See also Gosling Dir. ¶ 63 ("Microsoft uses its development tools to encourage developers to write Microsoft-dependent Java-based programs. Microsoft bundles its Java development toolkit in its Visual Studio development suite, which is the most popular set of Windows development tools. Microsoft's Java-based development tools are shipped with the company's incompatible extensions enabled by default."); Gosling Dir. ¶ 69 (explaining that because "of the Microsoft dependencies embedded in these toolkits, programs written using the toolkits can cause Java-based programs to be tied to the Microsoft implementation, undermining the cross-platform compatibility of the Java technology"); Gosling, 12/10/98pm, at 37:13 - 38:2 ("Microsoft engaged in changes to their compiler and their development tools such that the binaries that it produced . . . would not actually run on all the different VMs. The output of their compiler in a large fraction of cases would only run on the Windows platform and on the Windows Java virtual machine." Gosling explained that "the output of their Java compiler, if you follow their standard recommendations, what happened was you would end up with something that was not cross-platformed. It was targeted only at their operating system and their Java virtual machine."); Soyring, 11/18/98pm, at 49:23 - 51:9 (testifying that "use of compiler directives and key words which are nonstandard key words that have been added to the Java language").

320.3. Third, Microsoft declined to include the Java Remote Method Invocation (RMI) class library, as standard Java component, in its implementation of the Java Runtime Environment (JRE) distributed with Internet Explorer and Windows.

320.3.1. Microsoft refused to distribute RMI, a cross-platform communications protocol, with the JRE included in Internet Explorer 4 and thus Windows 95 and Windows 98.

i. Gosling explained that "RMI is a way for a Java program running in one place to send a message to a Java program in another place and have something happen, and maybe some communication comes back and forth. It's really a communication mechanism for two entities to just send messages back and forth." Gosling, 12/3/98pm, at 66:3-7.

ii. Soyring testified that Microsoft “deviates in several areas with what we understand the standard to be as proposed by Sun Microsystems, and for which case, we need to run test cases to insure the compatibility of our Java applications across different platforms. That . . . includes the omission in the Internet Explorer and Windows 98 product of a function called RMI, Remote Method Invocation, which is used in many of our applications." Soyring, 11/18/98pm, at 50:16 - 51:1.

320.3.2. Indeed, Microsoft made RMI available only from an obscure portion of its web site, where it was difficult for consumers and developers to find.

i. As an August 1997 e-mail from Microsoft’s James Van Eaton indicated, RMI was relegated to a relatively inaccessible location: "There will be no entry in the index for this file. They'll have to stumble across it to know it's there. <u>ftp://ftp.microsoft.com/ developr/msdn/unsup-ed</u> is where I put it on the Internet with 37 other old files in this directory. I'd say it's pretty buried." GX 1931 (emphasis in original).

ii. <u>See</u> <u>also</u> Gosling Dir. ¶ 58(c) (testifying that the “only way for a person using Microsoft’s tools or its JVM to take advantage of RMI functionality is for them to download and install the RMI class libraries and RMI compilers from the Internet. While some technically proficient and knowledgeable developers may be able to find these RMI technologies on the web, most end users with a Microsoft JVM will never know of their existence, let alone be able to find, download, and install them."); Soyring, 11/18/98pm, at 51:1-6 (“Granted, this piece of code, RMI, is available from Microsoft on their web site, but it’s very difficult to find, and our customers lack confidence that it would continue to be supported unless it’s in the base product, itself, as dictated in at least our understanding of the Sun specification for Java.”); DX 2025 (PC Magazine observed that although Microsoft “makes the RMI code available on its Web site, this is much less convenient to access than if it were included as part of the VM. Java programs that implement RMI won't work on the Microsoft environment as shipped.").

321. As a result of Microsoft’s refusal to support JNI and RMI, and its development and distribution of Windows-specific interfaces and extensions to Java development tool products, developers

were required to choose between compatibility with the Microsoft JVM (and thus Windows), on one hand, and compatibility with virtually all other JVMs (which could run on a variety of operating systems, including Windows), on the other hand. Applications that use Microsoft's interfaces do not run on JVMs other than Microsoft's or on operating systems other than Windows.

i. Gosling summarized the impact of the “choice” that Microsoft's refusal to support JNI and its simultaneous provision of Windows-specific native interfaces provides Java developers: "By omitting the standard API, Microsoft prevents developers from writing a Java-based program that can run on every JVM implementation. If the developer uses JNI, the software will not run on Microsoft's JVM; if the developer uses Microsoft's proprietary RNI, J/Direct, or @COM interfaces, the software will not run on any other JVM." Gosling Dir. ¶ 58.

ii. See also Gosling, 12/10/98pm, at 40:18 - 41:5 ("If you had used JNI to interface with native methods, you would find that your native method interface would not work at all with their virtual machine, and you would be faced with the task of reengineering their native methods to use one of the three Microsoft-provided native method technologies. And, of course, once you have gone through that work of reengineering your native method code to use the Microsoft native method code, you would find that you had landed in the trap because these native methods using J/Direct or COM or RNI don't run anywhere else. So, once you sort of slid in there, you have gotten yourself mired."); Gosling, 12/10/98pm, at 38:3-9 ("if you used some other development tool to develop a Java program, . . . essentially, any Java program that you developed using native methods using the standard development tools would not work on the . . . Microsoft virtual machine"); Gosling, 12/10/98am, at 23:20 - 24:13 ("The situation that Microsoft's actions put us in is one where developers, by and large, can end up having to test twice: once on a certified VM to see that they will work across, you know, Solaris, OS/2, Apple, or whatever, and then they separately have to test for Microsoft because of the choices that Microsoft has made.”).

iii. Soyring testified that Mirosoft's hiding of RMI leads to a “lack” of “confidence that it will continue to be supported,” resulting in “a reluctance to use RMI on other platforms because it would break the cross-platform compatibility goal they are trying to achieve.” Soyring, 11/18/98pm, at 89:4-13.

322. Microsoft, having developed its Windows-specific Java implementation bundled it with

Windows and Internet Explorer to ensure its wide-spread distribution.

i. A January 1996 Microsoft presentation describes as a "Response Summary" to cross-platform Java: "Increased IE share, Integrate with Windows." GX 52, at MS 003270.

ii. IBM recognized that "Internet Explorer is Microsoft's primary weapon to kill 100% Pure Java." DX 1894, at IBM 60968.

iii. Gosling testified: "Microsoft has used its ubiquitous operating system to flood the market with its Microsoft-dependent implementation . . . both in the version available at retail stores, and in the version it licenses to PC manufacturers to install in new PCs. These are powerful methods of distributing Microsoft-dependent JVMs, and they assure that Microsoft-dependent JVMs will be present on a very large percentage of personal computers." Gosling Dir. ¶¶ 54, 66.

iv. Increasing Internet Explorer's share, Robert Muglia agreed, was a response both to Netscape Navigator and to Sun's cross-platform threat in "the sense that the Internet Explorer feature of Windows has APIs in it that are competitive with Java." Muglia, 2/26/99pm, at 53:25 - 54:1.

v. Microsoft's Ben Slivka testified that, because of the enormous market share of Windows, shipping software with Windows could "create a de-facto standard." As such, Slivka considered distributing Microsoft's implementation of Windows as one of Microsoft's "levers for Java." Slivka Dep., 1/13/99pm, at 748:12 -749:6.

vi. <u>See</u> <u>also</u> Soyring Dir. ¶ 28 (Microsoft's tying arrangement also advantaged "Microsoft's implementation of Java technology").

323. By distributing its Windows-specific version of Java widely, and forcing developers employing it either to sacrifice the use of Windows code or Microsoft interfaces on the one hand, or compatibility with multiple Java virtual machines on the other, Microsoft encouraged a fragmentation of the Java platform that undermines the prospects for success of cross-platform Java.

i. IBM's John Soyring testified that customers "have expressed concern about Microsoft's implementation of Java"; specifically, that "it deviates in several areas with what we understand the standard to be as proposed by Sun Microsystems, and for which case, we need to run test cases to insure the compatibility of our Java applications across different platforms. That includes the admission from the Microsoft implementation of a function called JNI, which is an acronym standing for `Java Native Interface.'" Soyring, 11/18/98pm, at 49:23 - 50:23.

ii. As Sun's Graham Hamilton wrote: "We visited Microsoft today to hear the technical details of their plans to 'put java into Windows 95.'" "We asked how Java ISVs will get at all the wonderful Microsoft APIs from within Java. We used the new game APIs as an example. The answer is that Microsoft will either provide Java class wrappers to C interfaces, or maybe provide an automated way of accessing Win32 APIs via OLE/COM. This will be done on a case-by-case basis as needed and/or requested. Clearly this is a very disturbing answer. This basically means they will be encouraging Java ISVs to write to a broad set of APIs that are only available on Win32." DX 2053.

iii. Franklin Fisher summarized the impact of Microsoft's conduct with respect to native interfaces: "Microsoft took actions to impede the cross-platform potential of Java by developing an interface called J/Direct. Any application that uses 'J/Direct will run only on the Microsoft virtual machine.' The default way of writing applications and applets for Microsoft's virtual machine causes some of those applications and applets not to be able to run properly on non-Windows platforms or even on non-Microsoft virtual machines running on Windows." Fisher Dir. ¶ 208 (summarizing Microsoft documents).

iv. James Gosling testified: "By forcing computer manufacturers to bundle both a browser and Microsoft's JVM, Microsoft has hindered the ability of other browsers with compatible JVMs from getting distribution on Windows PCs." Gosling Dir. ¶¶ 68.

323A. Microsoft seeks to justify its strategy of polluting and fragmenting Java by arguing that this strategy "does not differ in a significant way from Sun's strategy." MPF ¶ 850. But Microsoft cites no evidence that Sun has undertaken to pollute or fragment Java by adding Sun-specific interfaces or extensions or by refusing to support cross-platform Java functions. Moreover, unlike Microsoft, Sun lacks the ability to fragment Java (even if it sought to do so) because it lacks Microsoft's large user base and monopoly power. See supra Part II.

2. Microsoft's purpose in polluting Java was to reduce the threat that cross-platform Java posed to the applications barrier to entry

324. The purpose of Microsoft's simultaneous exclusion of JNI and its development and

promotion of Windows-specific alternatives was to create incompatibility and “kill cross-platform Java,” not to enhance performance of Windows-specific programs.

i. An internal planning memorandum for Microsoft’s Java development tools for 1998 reaffirmed the objective: “Kill cross-platform Java by grow the polluted Java market.” GX 259.

ii. A fall 1996 presentation on “Microsoft API Strategy” used by Muglia summarized Microsoft’s “Java Approach” as including the following plan: “Use our Windows VM to own Java.* and Beans . . . . MS ability to lead Java developers is largely driven / limited by IE's share . . . . How does Microsoft win?” “Requires leadership in browser market share, developers target Microsoft API extensions." GX 470, at MS6 5006862, MS6 5006877, MS6 5006887 (emphasis added); see also Muglia, 2/26/99pm, at 58:4-23.

iii. Microsoft’s Todd Nielson summarized the strategic objective behind J/Direct in an e-mail to Gates and Chase on August 25, 1997: "So we are just proactively trying to put obstacles in Sun's path and get anyone that wants to write in Java to use J/Direct and target Windows directly." GX 256.

iv. In an August 1996 e-mail concerning “Java APIs,” Microsoft’s John Ludwig similarly emphasized: "Subversion has always been our best tactic. Don't fight awt--enhance it and support it better than anyone else. Don't fight jbdc--adopt it and move on. Subversion is almost invariably a better tactic than a frontal assault. It leaves the competition confused. They don't know what to shoot at anymore." GX 1327 (emphasis added).

v. Ben Slivka and others continued to emphasize the same theme in an October 1996 presentation articulating Microsoft’s "Java Strategy”: “1) Drive MS Java VM and classlibs (w/Win32 extensions!) to broad installed base . . . Do encourage fragmentation of the Java classlib space: multiple 2D standards, AWT vs. IFC, etc." GX 518.

vi. Steven McGeady testified that Microsoft’s Java strategy was “part of the embrace-extend model, which was, I believe, by this time they felt that it was unlikely that they could keep Java from happening at all, but they wanted to have it happen in a way that was incompatible. They wanted Microsoft's Windows Java to be incompatible with Sun Java. That would essentially have defeated the fact that someone could write a Java application and have it run on a cross-platform basis. If Sun's Java had one model and Microsoft's had a different model, then Java applications wouldn't work everywhere, and they wanted that to--they wanted to keep that from happening . . . . They wanted them--they wanted Java applications written for Windows to just run on Windows, and Java applications conversely written for non-Windows platforms to not run on Windows." McGeady, 11/9/98pm, at 71:23 - 72:18; see also McGeady, 11/10/98am, at 10:19 - 11:7 ("And whether or not it had any benefits for end users, it would have made the Java applications written for Windows incompatible with other platforms.").

vii. Gosling recalled Microsoft executives agreeing in a February 1997 meeting that fragmenting Java would harm developers: "All of us who attended the meeting -- including Microsoft -- unanimously agreed that unilaterally extending the Java programming language would hurt compatibility among Java tools and programs, would injure other tools vendors, and would damage customers' ability to run a Java-based software product on whatever platform they wished. One of the Microsoft representatives in attendance admitted that unilateral language extensions would be detrimental, and said that Microsoft 'wouldn't be cowboys' by unilaterally introducing such extensions in their implementation of the Java technology." Gosling Dir. ¶ 61.

viii. Slivka wrote in October 1996: "If Sun and we disagree on this and diverge in more areas, as long as Netscape doesn't buddy up with Sun, that is a super outcome for us (more fragmentation)." GX 1910. Slivka explained in his deposition that the "this" in "if Sun and we disagree on this and diverge in those areas" meant language extensions. He also acknowledged that it would be a super outcome, A. "Because it would just confuse Java developers about which Java platform they should write for. Q. That was a good thing from Microsoft's perspective? A. Correct." Slivka Dep., 1/13/99pm, at 753:17 - 754:7; See also Slivka Dep., 1/13/99pm, at 735:13-23 ("Q: And did you see the fragmentation of the Java class library space as being useful because it was a means to the end of defeating 'Write Once, Run Anywhere'? A: All this comes back to Windows and the threat, you know, Sun's very direct threat to our Windows platform, and the success of Windows on the client. So, this seemed like if the library space fragmented, the 'Write Once, Run Anywhere' . . . would be a lot less probable.").

324.1. Microsoft's own, contemporaneous documents show that its purpose was, not to increase the value of or demand for Windows, but rather to "kill cross-platform Java."

i. An internal planning memorandum for Microsoft's Java development tools for 1998 reaffirmed the objective: "Kill cross-platform Java by grow the polluted Java market." GX 259 (emphasis added).

ii. In October 1996, Slivka and other executives articulated "Microsoft Java

Strategy" as including: "Let Java class library space fragment so that 'write once, run anywhere' does not happen . . . . Don't encourage new, cross-platform Java classes, especially don't help get great Win32 implementations written/deployed." GX 518 (emphasis added).

iii. In May 1997, Gates sent an e-mail to John Ludwig and Ben Slivka stating, "I think supporting JDK 1.1 is fine and I am hard core about NOT supporting JDK 1.2. I really needed to understand where we were going to draw the line because I am so afraid of the slippery slope." Slivka responded: "JDK 1.2 has JFC, which we're going to be pissing on at every opportunity. This summer we're going to totally divorce Sun. Apple's announcement yesterday that they will encourage Rhapsody developers to write in Java and call native Rhapsody services was a mere shadow of what we're going to be encouraging developers to do." GX 253.

iv. In January 1997, Gates sent an e-mail to Maritz and other top Microsoft executives in which he commented on Java as follows: "To avoid middleware taking over an operating system you have to make sure the integrated services are different from the middleware — otherwise the middleware approach has no disadvantages and it wins. I think the path we were going down of building on AWT was a sure disaster — it was creating a situation where pure 100% Java applications would look just as good as pure Windows applications which we have to avoid." GX 351.

v. In April 1997, Jim Allchin wrote: "The comment about AFC concerns me since I think you are saying that whatever functionality is added by SUN you will add in a compatible way. They have you on a treadmill. I don't understand how this is a winning course. I would explicitly be different — just to be different. Further, I believe the highest priority is to create and expose NEW Windows APIs (not clone) — absolutely callable from java. I would add more to Windows and less cross platform. This is clearly a hard tradeoff that will be made each day, but I believe it is not a question of how much resource is applied between Windows and cross platform." He continued that "without something to pollute Java more to Windows (show new cool features that are only in Windows) we expose ourselves to more portable code on other platforms — especially if the Pure Java logo push wins in any way. Either we need to pollute to Windows or we need more other languages to offer viable alternatives to Java (or both). I would feel so much better if I felt we weren't on such a middleware clone SUN strategy. This is where" you and "I are apart." GX 1334, at MSS 0003551 (emphasis added).

vi. Fisher Dir. ¶¶ 111-113 (collecting quotes and documents); Fisher 1/7/99pm, at 50:11-19 (commenting on GX 518: "well, what Microsoft did with Java, as I testified this afternoon, was essentially to engage in a strategy which discouraged or subverted cross-platform Java. And this is quite consistent with it. Don't encourage cross-platform Java classes, and it would doubtless help the Java if applications written in cross-platform Java had great Win32 implementations. Win32, as I assume we all know, is 32-bit Windows, which refers to Windows 95 and 98 and Windows NT.”).

vii. See also GX 58 (Slivka April 1997 list of "issues/concerns" to Mr. Gates, stating that "I want to make sure I understand your issues/concerns. Here's a start, can you please add anything I'm missing?" He questioned "2. How do we wrest control of Java away from Sun? 3. How do we turn Java into just the latest, best way to write Windows applications?").

324.2. If Microsoft had wanted to increase the value of Windows, it would have supported JNI; supporting JNI would have reduced development costs for Windows programs because it would have enabled developers to write simultaneously for both Windows and other platforms.

i. As Maritz described Intel's perception of Netscape in April 1996, "they see Netscape as a 'friend of the PC'" (GX 288) suggesting that ensuring compatibility with Navigator's Java implementation would promote PC-based (and therefore Windows-based) application development.

ii. Gosling testified that it was unnecessary to exclude JNI in order to achieve the goal of providing developers the ability to develop Windows-specific Java programs: "We believe that it is perfectly fine for them to be allowed to make a choice between building a native method in a portable way and building a native method in some other way. In our case in San Jose, we never tried to represent that we thought it was a bad idea for Microsoft to put their own native method mechanism in there. What we argued was that it was bad for them to exclude the portable native method interface that was a part of the standard. And their argument had been that it was difficult, but it has only taken them a few weeks to actually release a new version of the tool. So it really wasn't all that difficult for them to do." Gosling, 12/10/98am, at 54:8-20; see also Gosling, 12/10/98am, at 33:2-14 ("Microsoft could easily have accomplished its goals of building an access to the underlying Windows platform, without violating the Java standards--without violating the Java specification. In fact, as the system was initially delivered to Microsoft, there was already a native method interface . . . called 'NMI' which worked perfectly well. It had a number of issues, as far as developers were concerned, with portability amongst different platforms, but, nonetheless, it allowed people to build native methods that would access whatever underlying platform facilities people needed to expose. And that was all done within the context of the Java standard.").

324.2A. Microsoft now concedes that J/Direct is and was intended to be Windows-specific. See MPF ¶ 849. This admission, however, is inconsistent with Microsoft witnesses' attempts to argue that J/Direct was not intended to be Windows-specific and thus illustrates the implausibility of its witnesses' effort to justify Microsoft's Java conduct.

i. See, e.g., Gates Dep. (played 12/2/98am), at 27:22 - 28:1 (Q: Why was J/Direct developed by Microsoft? A: To make it easy for people who choose the Java language to call the unique runtime features in various operating systems, including Windows."). See also Muglia, 2/26/99pm, at 82:13-16 ("I'm not aware of any other operating systems that support J/Direct, although I do know that we openly publish the specifications so that others could take advantage of it."). As Dr. Gosling pointed out (12/10/98am, at 27-28), making Microsoft's native interfaces available to work with other operating systems is "not a trivial engineering effort" inasmuch as "there is essentially no way to implement them without reimplementing some significant part of Windows on whatever your system happens to be."

324.3. Indeed, no other JVM vendor or operating system vendor has taken steps to support Microsoft's Windows-specific native code interfaces, and most have supported JNI.

i. Gosling testified: "So, for doing native method interfaces outside of browsers, this has worked quite well, and JNI has been fairly healthy. As far as I know, essentially all of our licensees have implemented JNI, although they're not, by and large, browser vendors." Gosling, 12/9/98pm, at 97:24 - 98:6; see also Gosling, 12/9/98pm, at 98:18-20 ("As I understand it, essentially all of our licensees are in pretty good shape when it comes to native methods.").

324.3A. In light of the fact that supporting JNI would enhance the value of Windows, there is no legitimate business reason -- other than protecting Microsoft's

operating system monopoly -- for Microsoft's failure to do so. Microsoft's JVM can support multiple interfaces, and Microsoft's compliance with the preliminary injunction entered by the District Court in November 1998 in Sun Microsystems, Inc. v. Microsoft Corporation demonstrates that it can readily support JNI.

i. As David Bowen of Sun indicated in September 1996, "I believe Tim and Cheng have been talking to someone from MS and believe that it will not be hard for them to support our new proposed API on top of their COM interface." DX 1918.

ii. Similarly, the Sun Court found that a "ninety-day period, with the right to extend for good cause, should allow Microsoft adequate time to modify its virtual machine to support JNI." Sun Microsystems, Inc. v. Microsoft Corporation, 21 F. Supp.2d 1109, 1126 (N.D. Cal. 1998).

324.4. Microsoft's argument that it developed its own native code interfaces and refused to support JNI because it was excluded from the Java standard-setting process is not consistent with the evidence.

324.4.1. Microsoft was never interested in promoting cross-platform Java.

i. In February 1997, Ben Slivka and other Microsoft employees discussed "potential issues for Java Language meeting." Paul Gross counseled against such a meeting, noting, "I believe that our true goal, controlling the future of Java, will be totally transparent and mostly unacceptable to all Java OEMs. I believe that we currently have the most control and at some point will have total control." GX 1326.

ii. Gosling testified that Microsoft's communications concerning native interfaces indicated "that Microsoft was completely uninterested in pursuing a technical investigation of a portable solution." Gosling, 12/9/98pm, at 57:12-25; see also Gosling, 12/9/98pm, at 67:1-8 (testifying that "when the time came to actually form a working group, since Microsoft had already said they weren't interested in doing anything that was a cross-platform design, there was not a lot of point"); Gosling, 12/10/98pm, at 52:24 - 53:12 ("their version of

> friendship was just take what we give you, and all of these things are essentially poison pills that break cross-platform portability . . . the proposals that they made were fairly consistently things which tied people into the Windows platform in ways that broke all of the interoperability goals.").

324.4.2. Mr. Muglia's testimony (Muglia Dir. ¶¶ 61-69, 108), that Microsoft attempted unsuccessfully to work with Sun on native code interfaces and that Sun's internal documents somehow suggest a plan unfairly to exclude Microsoft, rests entirely on selected documents furnished by Microsoft counsel for the purposes of litigation and on statements in these documents that Mr. Muglia read out of context to imply the opposite of their intended meaning.

i. Muglia acknowledged that his argument was based on a selection of Sun documents provided to him by Microsoft's attorneys, not any independent or comprehensive review. Muglia, 2/26/99pm, at 90:3-19.

ii. Muglia relied in part on a series of September 1996 Sun e-mails titled "Java Interfaces and Beans" (DX 1918) to support his suggestion that Sun simply "ignored" Microsoft's input (Muglia Dir. ¶ 67). But the Sun e-mails do not suggest any unfair exclusion. John Kannegaard initiated the exchange by informing other Sun executives that "MS grumpy that we aren't just going with their specs. We should tell them why." In response, David Bowen noted that although "we are not going to adopt the MS solution," Sun engineers had been working with Microsoft and, based on that work, "believe that it will not be hard for them to support our new proposed API on top of their COM interface." Similarly, David Connelly noted: "Consistently, Microsoft's idea of working with us is to offer us some specs clearly aimed at the Windows 32 platform, then balk when we don't just accept them. This was clearly true for JAR and, perhaps almost everything else on the list." DX 1918.

iii. Muglia testified that Sun executives had recognized and expressed concerns about limitations on Sun's rights to insist on Microsoft's compliance with its specifications under its Technology License and Distribution Agreement (TLDA), cited internal Sun e-mails (DX 1956) in support of this, and suggested that top Sun executive Jon Kannegaard "concurred that the TLDA posed problems for Sun and solicited

approaches to those problems." Muglia Dir.
¶ 68. Muglia ignored, however, that Kannegaard explained in no uncertain terms in his January 1998 deposition in Sun Microsystems, Inc. v. Microsoft Corp., that the document contained a typographical error and that in fact "I did not reach such dire conclusions" as those expressed in DX 1956. Kannegaard Dep., 1/19/97, 184:1-2 (DX 1956A) (sealed).

iv. Muglia also relied on DX 2174, yet another series of internal Sun e-mails, to support his contention that Sun deliberately excluded Microsoft. Muglia Dir. ¶ 68. He ignored, however, that Kannegaard specifically testified in his deposition that Sun never undertook any of the actions discussed in DX 2174: "This is really an amazing e-mail message, isn't it? We didn't do that." See also Kannegaard Dep., 1/19/97, at 195:1-4 (DX 2174).

v. Sun documents also contradict Muglia. See GX 1357 (Kannegaard e-mail messages to Microsoft, 10/2/96: "I looked into complaints you've heard and relayed from microsoft on how we ignore them, surprise them etc. In particular, they complained about the JRI design, a design that did not go their way. We conducted a fabulous process and came to the right conclusion. They just don't like it." "You can look at this yourself. If you do, you will find that the Microsoft proposal was seriously considered but, in the end, rejected for legitimate technical reasons. You will find that IBM, Apple, and SMLI all reached this conclusion without any help from us.").

324.5. Similarly, Microsoft's refusal to distribute RMI, as well as its making RMI difficult to find on its website, made no business sense except as part of Microsoft's campaign to kill cross-platform Java and maintain its monopoly power.

324.5.1. Microsoft's refusal to distribute RMI increased Microsoft's costs.

i. Gosling testified: "In packaging it separately, . . . they did incur an extra cost for themselves because they had to create the separate package rather than just delivering them together. RMI is a very small fraction of the size of the base tool, but what it has done for developers is it's made using this cross-platform communication technology much more awkward because then their customers have to go through this other step of installing this patch to actually make it work." He continued: "And what they effectively did was create a separate product. A

> separate set of binaries that you could download separately. And it would have been a pretty straightforward thing, and from my point of view, a very natural thing to have left the RMI classes with everything else. It certainly wouldn't have made the product any larger. It wouldn't have changed their testing problem at all because they had to test the separate product in combination with this, anyhow. It was simply a decision to package it separately and to incur the costs of building two packages instead of one." Gosling, 12/10/98pm, at 41:12 - 42:24.

324.5.2. Microsoft's contention that Microsoft "elected not to ship RMI with our Java implementation because we think it is inferior technology" (Muglia Dir. ¶ 113) is unsupported by any contemporaneous evidence.

i. Gosling, 12/10/98pm, at 42:14-24 ("Q: What advantage, if any, was there to Microsoft in doing that? A: It certainly made developers' lives harder. Q: Any other advantage? A: Well, they had a set of communication technologies, themselves, which they have been trying to promote which are, you know, not standard cross-platform communication technologies. And they, in some ways, conflict with RMI, but that's a business conflict. It's not a technical conflict. It's perfectly possible to run DCOM and RMI together.").

325. Microsoft contends that its pollution and fragmentation of Java were intended merely to provide developers the choice of writing superior Windows-specific Java programs, without harming the prospects of cross-platform Java (Muglia, 2/26/99pm, at 27:23 - 28:24, 36:24 - 37:8; Schmalensee Dir. ¶ 149). But this assertion is also inconsistent with the evidence.

325.1. <u>First</u>, Microsoft's professed commitment to improving the Java programming environment is belied by contemporaneous documents showing that its intent was to "subvert" and "kill" cross-platform Java.

i. As its top executives discussed in contemporaneous e-mails, Microsoft wanted its Java implementation to "be different just to be different" (GX 1334, at MSS

0003551 (Allchin)) and to “subvert” Java (GX 1327 (Ludwig)). See generally infra VI.A.3.b.(1); ¶ 330.

ii. As Gosling explained in response to the Court’s questions about Microsoft’s suggestion that it has simply improved Java, that Microsoft’s “version of better is tied to the Windows platform and preventing interoperability with other platforms.” Gosling, 12/10/98pm, at 62:3 - 63:24.

iii. Similarly, Soyring testified that “I can only comment on what I read in the press and reading Microsoft press releases, is that they do talk frequently about their support of Java. They talk about having the highest performance Java implementation, though it’s a Microsoft specific version of Java. They talk about providing Java application development tools, which is the Microsoft Visual J++. So, they use those very frequently as evidence that they are an active proponent of Java. It just happens to be misleading, and many of our customers recognize that because of the proprietary extensions or deletions or omissions that are in their Java implementation.” Soyring, 11/18/98pm, at 55:17 - 56:8.

iv. A March 1998 unreleased draft of a Symantec public statement put Microsoft’s illusory creation of “choice” succintly: “By breaking away from pure Java, there is a perception that MS is seeking to cloud the issue; throwing up new whiz-bang technologies which have little practical use in 90% of applications.” GX 2078.

v. Ben Slivka stated in an October 1996 e-mail: “If Sun and we disagree on this and diverge in more areas, as long as Netscape doesn’t buddy up with Sun, that is a super outcome for us (more fragmentation).” GX 1910. He elaborated on this comment in his January 1998 deposition: “Q: The last paragraph you say, "if Sun and we disagree on this and diverge in those areas"--first of all, by "this," do you mean language extensions? A: Yes. Q: "If Sun and we disagree on this and diverge in these areas, as long as Netscape doesn't buddy up with Sun, that is a super outcome for us (more fragmentation).” Why would that be a super outcome? A: Because it would just confuse Java developers about which Java platform they should write for. Q: That was a good thing from Microsoft's perspective? A: Correct.” Slivka Dep., 1/13/99, at 753:17 - 754:7; see also id. at 735:13-24 (“Q: And did you see the fragmentation of the Java class library space as being useful because it was a means to the end of defeating ‘Write Once, Run Anywhere’? A: All this comes back to Windows and the threat, you know, Sun’s very direct threat to

> **our Windows platform, and the success of Windows on the client. So, this seemed like if the library space fragmented, the 'Write Once, Run Anywhere' . . . would be a lot less probable.").**

325.2. Second, Microsoft's contention that it was simply providing developers and consumers options is belied by the numerous steps it took to deny consumers Java implementations not controlled by Microsoft, even when those implementations would run well on Windows.

i. McGeady testified that "Microsoft was very upset that we were working on Java at all, and in particular, they didn't want -- we had a very highly optimized virtual machine that ran very fast on Intel architecture, and they did not want that VM to become part of Netscape Navigator." McGeady, 11/10/98am, at 6:9-16; see also McGeady, 11/10/98am, at 10:19 - 11:7.

ii. Engstrom made clear to Intel that Microsoft would prefer it to stop helping Sun on Java APIs, "especially ones that run well . . . on Windows." GX 235.

iii. Microsoft's Todd Nielson -- the executive in charge of Microsoft's Developer Relations group -- summarized Microsoft's strategic objective in August 1997 as "just proactively trying to put obstacles in Sun's path and get anyone that wants to write in Java to use J/Direct and target Windows directly." GX 256.

iv. As David Cole put it in a July 1997 e-mail: "We've agreed that we must allow ISVs to redistribute the Java VM standalone, without IE. ISVs that do this are bound into Windows, because that's the only place the VM works, and it keeps them away from Sun's APIs." GX 728.

325A. Microsoft's pretextual suggestions that its fragmentation of Java has been motivated by, and has had, the procompetitive effect of promoting choice among Java implementations (MPF ¶ 833) is inconsistent with its argument elsewhere that fragmentation of the Windows platform reduces its value to users, developers, and OEMs. MPF ¶¶ 164; 544-548.

i. Compare Muglia, 2/26/99pm, at 36:24 - 39:21 ("Q: Now, my question is regardless of where the fragmentation of Java came from, did Microsoft

believe that it was in Microsoft's interest to have that fragmentation occur? A: Yes, in essence that's true. In this case, remember I was saying fragmentation is really a word, and particularly with Java, meaning choice, different choices for developers. And it is in our interest to have more choices for developers, yes."), with*:*

ii. **MPF ¶ 548 ("Widespread modifications to Windows that undermined it as a stable development platform would likely retard the growth of the personal computer industry generally, thus injuring all OEMs as well as Microsoft. (Kempin ¶ 34.) Microsoft should not be forced to facilitate the fragmentation of Windows by permitting OEMs to modify Windows however they please. (Kempin ¶¶ 33-34.)").**

326. The testimony of Microsoft witnesses concerning its conduct relating to Java is not credible.

326.1. Mr. Gates' testimony is contradicted by the contemporaneous documents.

i. Compare Gates Dep. (played 12/2/98am), at 20:14-18 ("Q: Is one of the things that you're focused on trying, in Mr. Slivka's words, to wrest control or get control, if wrest is a word that you don't like, of Java away from Sun? A: No."), with GX 265 (Gates, August 1997: "Do we have a clear plan on what we want Apple to do to undermine Sun?"); GX 897 (Gates, January 1998: "There is a big question of what we should do with JAVA on Mac — whether doing more work and working closely with Apple could help us in this 'battle'. I think it can and would hate for Apple to have to go back to the SUN camp.").

ii. Compare Gates Dep. (played 12/2/98am), at 21:11-16 ("Q: It is your testimony, then, sitting here, that Microsoft was not at all motivated by a desire to limit the extent to which Java could be used to develop applications programming that could be used on platforms other than Microsoft's Windows? Is that your testimony? A: Yes."), with GX 290 (Gates February 1997 message concerning Microsoft support for AMD's 3DX work: "If Intel has a real problem with us supporting this then they will have to stop supporting Java Multimedia the way they are. I would gladly give up supporting this if they would back off from their work on JAVA which is terrible for Intel.").

iii. Compare (Gates Dep. played 12/2/98am), at 32:2 - 33:21 (testifying, regarding a document written by Slivka, that "I don't know if he's referring to pissing on

JFC or pissing on JDK 1.2, nor do I specifically know what he specifically means by "pissing on." And testifying that “he might mean that we're going to be clear that we're not involved with it, that we think there's is a better approach”), with GX 253 (Gates’ e-mail message to which Slivka was responding: “I am hard core about NOT supporting JDK 1.2.”).

326.2. Mr. Muglia’s testimony that Microsoft wanted to support Sun’s version of Java and not fragment Java is contradicted by Microsoft’s own documents and is internally inconsistent.

i. Muglia’s effort to explain away Mr. Gates’ statements that “I am hard core about NOT supporting JDK 1.2" (GX 253) by testifying that "exactly what Bill meant by the word 'support' isn't really clear" is incredible. As the Court noted, "I don't think it could be read any other way" than to make clear that Gates "doesn't like the idea of supporting" JDK 1.2. Muglia, 2/26/99pm, at 43:23 - 48:10. Muglia’s subsequent effort to further explain that Microsoft drew a distinction between "supporting" JDK 1.2 on its website, while not distributing it with Windows (Muglia, 2/26/99pm, at 110:7 - 111:6), implicitly conceded that GX 253 accurately stated Microsoft’s policy of seeking to impede cross-platform Java.

ii. Although Muglia suggested that “killing cross-platform Java,” as expressed in GX 259, was merely the erroneous impression of a low-level employee and not Microsoft’s strategy (Muglia, 2/26/99pm, at 99:9 - 100:3), that testimony is belied by numerous internal documents and Muglia’s own testimony. See, e.g., GX 351 (Gates: “I think the path we were going down of building on AWT was a sure disaster — it was creating a situation where pure 100% Java applications would look just as good as pure Windows applications which we have to avoid.”); GX 518 (Slivka: “Let Java class library space fragment, so that ‘write once, run anywhere’ does not happen”); Muglia Dir. ¶ 72 (“By encouraging developers to write Java programs that make native calls to Windows APIs, Microsoft is growing the market for Windows-specific, or ‘polluted’ Java. If most developers choose to write Windows-specific Java programs, instead of slower, less functional cross-platform Java programs, we will have won the competitive battle and ‘killed’ Sun’s strategy of using cross-platform Java to ‘attack’ Windows.”).

iii. Muglia also sought to explain away GX 1334 (Allchin, “we should be different - just to be different”) by arguing that it concerned Microsoft’s Application Foundation Classes (AFC), which Microsoft abortively sought to develop as a cross-platform alternative to Java, ignoring that this was a distinction without any meaningful difference: "Q: And do you understand that what he is saying is that he is recommending to Mr. Slivka that he not try to proceed by adding functionality in a compatible way, that he should be different just to be different? A: No, not exactly. Again, you have to understand the context of what this paragraph has in it. It says AFC. This was technology Microsoft was building for our customers, and I -- as I interpret this, Jim is saying, 'Don't focus on just the things that Sun is doing. If you do that, they have us on a treadmill. We need to do things that are different.' And he uses the emphasis phrase 'just to be different.' And as I interpreted that when I read it, it meant so that we can have differentiation versus our competitor, Sun. In other words, not just providing the same old thing that our competitor is providing. We need to be different. THE COURT: Why is that different that what Mr. Boies just said? A: Well, Mr. Boies wasn't referring -- I want to be clear, your Honor, that what Mr. Allchin was saying when he said this was in reference to one of Microsoft's technologies known as AFC." Muglia, 2/26/99pm, at 33:24 - 35:3.

iv. See also Muglia, 2/26/99pm, at 36:24 - 39:21 ("Q: Now, my question is regardless of where the fragmentation of Java came from, did Microsoft believe that it was in Microsoft's interest to have that fragmentation occur? A: Yes, in essence that's true. In this case, remember I was saying fragmentation is really a word, and particularly with Java, meaning choice, different choices for developers. And it is in our interest to have more choices for developers, yes. Q: Well, sir, are you saying that the reason that you wanted fragmentation was to just give developers more choice? Is that your testimony? A: No, not entirely. Q: Okay. Part of the reason that you wanted to have fragmentation of the Java platform is because that would discourage developers from writing to the cross-platform Java technology; correct, sir? A: Potentially, yes. Particularly if, in fact, the fragmentation is coming from our competitors, say, Sun and Netscape, each promoting a different variation of Java. If there were different implementations out there, yes, that might discourage developers from choosing our competitor's platform. Instead they might prefer to choose our platform, Windows.”).

3. Microsoft engaged in anticompetitive conduct to exclude cross-platform Java

327. Microsoft not only developed and widely disseminated its polluted version of Java, but also engaged in anticompetitive conduct to promote the use of its JVM and to exclude cross-platform

versions.

a. Microsoft used predatory means to weaken the principal distribution vehicle for cross-platform Java, Netscape

328. Microsoft impeded cross-platform Java by severely weakening Netscape, the principal distribution vehicle for cross-platform JVMs.

328.1. Microsoft's predatory campaign to weaken Netscape, and to gain browser share, hindered the distribution and usage of non-Microsoft JVMs.

i. See supra Part V.

ii. Gosling testified that, to the extent that Microsoft "did a fair amount of economic damage to Netscape, I mean, by essentially setting the price to zero, our biggest source of damage in Netscape as a distribution channel was when they got to the point where they couldn't fund the engineering efforts adequately to keep up with us." Gosling, 12/10/98pm, at 66:16-21; see also Gosling, 12/3/98am, at 50-51 (although Netscape has decided to stop shipping a VM with Communicator, this was a result of the fact that it just couldn't keep up with the engineering work of keeping their VM up to date, as a result of which it has decided to redesign the browser in a way that will work with separately distributed third party VMs).

iii. Gosling further testified: "I understand that evidence may be offered in this case that will show that Microsoft has taken actions that have limited the distribution of Netscape's browser. To the extent that Microsoft has taken such actions, those actions have made, and will continue to make, it more difficult for the Java technology to achieve the broad distribution necessary to ensure its success as a cross-platform technology. The preservation of a competitive market for Internet browsers is critically important to preserving the viability of the cross-platform Java technology. Microsoft's actions injure not only Sun, but the entire community of Java-based developers, computer users, and potential operating system competitors of Microsoft. I believe that a competitive browser market would significantly assist Sun and other cross-platform Java-based developers in getting cross-platform Java technology distributed to end users through the distribution of non-Microsoft Internet browsers . . . . The ultimate success of the Java technology is dependent on broad distribution of JVMs that comply with the specifications for the Java technology. Without enough users able to run Java programs, there will be little incentive for developers to write Java-based software; without enough Java-based software, the Java technology will be unattractive to users." Gosling Dir. ¶¶ 45, 64-65; see also Gosling, 12/3/98pm, at 32:10-20 ("We still believe that the browser is an incredibly important channel to get compatible Java virtual machines out there. And just because Netscape was unable to do the engineering to deal with the compatible Java virtual machine, that left us in something of a pickle. So, we decided that it was important because that channel was still very important, to try to work with them, despite their difficulties, . . . on a way where they could do less work and yet still have a . . . compatible Java virtual machine integrated effectively in their browser."); Gosling, 12/10/98pm, at 32:18-33:7 (testifying that, because of the difficulty of downloading JVMs, impeding Netscape "pretty severely restricted the channel for us").

iv. Gosling testified that "the overwhelming majority of JVMs that have been distributed to end users to date were distributed with Internet browsers, specifically Netscape's Navigator/ Communicator, and Microsoft's Internet Explorer. No other distribution mechanism for JVMs has reached even a small percentage of the distribution achieved through the distribution of Internet browsers." Gosling Dir. ¶ 42.

v. See also Gosling, 12/10/98pm, at 31:14-23 (Navigator provided a means to get the Java runtime environment out to consumers, "so that the Java APIs are now available everywhere that Navigator is, and so people writing Java applications can write them and run them on those platforms"); Gosling, 12/10/98am, at 68:25 - 69:3; Gosling, 12/10/98am, at 71:25 - 72:5; Gosling, 12/10/98am, at 77:15-17 (the web "as a channel, is just absolutely drowned by the fact that the Microsoft VM is bundled in the vast majority of all computers built today."); GX 254 (June 1997 e-mail from Gates to Maritz).

vi. Professor Franklin Fisher testified: "Microsoft, recognizing that Netscape's browser was the primary distribution method for Java, sought to eliminate Java by eliminating Netscape's browser as a viable alternative." Fisher Dir.

¶207 (collecting Microsoft documents).

328.2. Microsoft intended its predatory campaign against Netscape and other browsers to hinder cross-platform Java.

i. A January 1996 Microsoft presentation made clear that the "integration" of

Internet Explorer with Windows was a crucial element of a strategy for responding to Java: "Response Summary: Increase IE share, Integrate with Windows." GX 52, at MS7 003270.

ii. Robert Muglia agreed that "Increasing Internet Explorer's share is a response both to Netscape Navigator and to Sun's cross-platform threat" "in the sense that the Internet Explorer feature of Windows has APIs in it that are competitive with Java," meaning Microsoft's JVM. Muglia, 2/26/99pm, at 53:1 - 54:7.

iii. As Microsoft's John Ludwig pointed out in a September 1996 e-mail concerning Microsoft's relationship with Sun: "The only thing that will bring them to the table is our continued success at growing IE share, and growing our VM share thru IE and Windows distribution, and leveraging that to create a successful position in Java class libraries." GX 2519.

iv. An August 1997 memorandum from IBM's John Thompson to Sun's Scott McNealy, Netscape's James Barksdale, and others, emphasized that widespread distribution and usage of Internet Explorer was of serious concern to the future of an "open" Java platform: "We must clearly explain to our customers that Internet Explorer is Microsoft's primary weapon to kill 100% Pure Java. Customers who want an open Java platform must vote with their browsers!" DX 1894, at IBM 60968.

v. As an executive with the investment capital firm Kleiner, Perkins, Byers & Caulfield informed Netscape after a meeting with Microsoft: "M/S supports Java the programming language, but is directly opposed to Java the 'OS'. . . . Shipping IE 3.0 with every Win95 desktop from Q4 1996 on will drive leadership for IE and ActiveX against Navigator and Java." GX 1233.

vi. See also Soyring Dir. ¶ 28 (testifying that an "additional effect of Microsoft's including Internet Explorer with Windows is the advantage it gives Microsoft's implementation of Java technology"); Fisher Dir. ¶ 209 ("Microsoft did not seek to 'kill cross-platform Java' merely by developing its own version of Java and marketing it on the merits in competition with

cross-platform Java. Instead, Microsoft used its monopoly powers over PC operating systems to induce and require industry participants to accept J/Direct (and IE which included J/Direct) instead of Java (and Netscape's browser which included Java).").

b. Microsoft used its monopoly power to force widespread

distribution and usage of its Windows-specific version of Java

329. Microsoft did not simply use anticompetitive means to hinder cross-platform Java's distribution; Microsoft also used its operating system monopoly to coerce firms to distribute and support its Windows-specific Java environment.

(1) Microsoft entered into exclusionary "First Wave" agreements with ISVs

330. Microsoft entered into dozens of exclusionary "First Wave" agreements with ISVs with the purpose and effect of hindering cross-platform Java. These agreements conditioned early access to information about Windows 98 and Windows NT -- information of vital competitive significance to ISVs -- on ISVs' agreement to make Microsoft's Windows-specific JVM the default JVM for any of the ISVs' Java programs.

i. GX 970, at RSC 618 (Rational Software -- DevlinF); GX 2071, at DOJ0000003 (sealed) (Symantec -- Eubanks).

. GX 2400-2497 (all sealed) (additional First Wave agreements).

ii. Gordan Eubanks, moreover, acknowledged that "early access to select specifications and beta releases of Windows NT 5 and SDKs on an ongoing basis via CDs and secure FTP site," both of which are conditioned in the First Wave agreement on Symantec's agreement to make Microsoft's JVM its default JVM, is both "important" and "necessary" to Symantec. Eubanks, 6/16/99pm, at 16:4- 17:19.

iii. Soyring testified to the nature and effect of Microsoft's agreements: "A second aspect is from a commercial software developer program, commercial software vendors who want to develop and sell an application that carries the logo or the mark design for Microsoft Windows 95 and Windows NT. In that contract, I understand, is the provision that requires that the vendor, if they use Java as a language, write it to the

Microsoft Java and ship the Microsoft Java virtual machine with their application. That poses, then, a fear amongst our customers that that Java application may only run on these Microsoft platforms and, thus, not be cross-platform capable as advertised by the other Java components." Soyring, 11/18/98pm, at 53:21 - 54:7.

iv. **See also Sun Microsystems, Inc. v. Microsoft Corporation, __ F.3d ___, 1999 WL 635783 (9th Cir. 1999) ("Microsoft correctly argues that the literal terms of its contracts merely required its licensees to use its virtual machine and did not forbid them from using Sun's as well. We agree with Sun, however, that because Microsoft's virtual machine is incompatible with Sun's standard, no licensee would ever use both, and so Microsoft's policy is tantamount to requiring exclusive use.").**

330.1. Microsoft was enjoined by a Federal court from enforcing those agreements in 1998.

i. Sun Microsystems, Inc. v. Microsoft Corp., 21 F. Supp. 2d 1109, 1127-28 (N.D. Cal. 1998) (enjoining Microsoft from conditioning "the right to use the 'Designed for Windows95(98)/NT' logo on the exclusive distribution of Microsoft's Java virtual machine").

330.2. Neither any witness nor any contemporaneous evidence describes any legitimate business purpose for the restrictions included in the First Wave agreements

(2) Microsoft misled developers into using proprietary extensions to develop Windows-specific programs

331. Microsoft misled developers so that they unwittingly wrote Java programs that turned out to be Windows-specific, even when the developer intended to create a cross-platform program.

331.1. Until ordered to do so by a federal court in 1998, Microsoft did not warn developers that using its extensions would bind their Java programs to Windows.

i. Gosling testified that "it's my understanding that there is absolutely no warnings in it about, you know, the fact that you're generating platform-dependent code, and I don't even know a way to turn it off using that tool." Gosling, 12/10/98pm, at 49:12-15; see also Gosling, 12/10/98am, at 62-63 ("if your sole source of information was the Microsoft product, you would not understand any of this" because the disablers are buried several menus deep); Gosling, 12/10/98pm, at 48:22 - 49:4 (describing VJ6 "I'm not aware of any warnings that appear in the documentation. When you actually run the application, if you are sufficiently clever, you can actually find some switches you can pull which say do or do not use the Microsoft extensions. But even those don't explicitly say, you know, . . . 'if you don't do this, you won't be portable.'").

ii. As Professor Fisher explained: "The problem is, if you do this in Visual J++, at least until quite recently, you would not be warned of a rather salient fact, which is, if you do that, you would be writing in a way that would make . . . the program you were developing runnable okay on Microsoft's JVM for Windows, but it would not permit you to port the program easily to other . . . platforms." Fisher, 1/7/99pm, at 28:20 - 29:5.

iii. Sun Microsystems, Inc. v. Microsoft Corp., 21 F. Supp. 2d 1109, 1127-28 (N.D. Cal. 1998) (enjoining Microsoft from: "Selling or distributing, directly or indirectly, any software development tool or product containing or implementing computer program code copied or derived from any Sun copyrighted program code for the Java Technology; as that term is defined in the TLDA, including SDKJ 2.0, SDKJ 3.0 and VJ 6.0, ninety (90) days after the date of this order unless such product: . . . (2) has the default mode in the compiler configured such that (a) Microsoft's keyword extensions and compiler directives are disabled and (b) has the compiler mode switch such that it enables, rather than disables, such keyword extensions and compiler directives, and (3) includes a warning which appears when a user elects to use the extended mode of the compiler (either when the user accesses the compiler from a DOS command line or when the user checks a box provided during execution of the compiler software) and which warns the user (a) that use of Microsoft's language extensions will result in compiled code which may not run on all compatible virtual machines, and (b) that future versions of Microsoft's development tools may be prohibited by court order from incorporating keyword extensions and compiler directives not contained in Sun's Java Language Specification; however, nothing in this order prevents Microsoft from removing the mode switch, keyword extensions and compiler directives from its Java software development tools and distributing or selling such resulting implementations.").

331.2. The absence of warnings about the impact of utilizing Microsoft's extensions created a significant potential for developer confusion and the creation by them of Java programs that were not cross-platform, but rather Windows-specific.

i. Gosling testified: “You start with they’re essentially a monopoly in the operating system, and there is the fact that they bundle their virtual machine . . . with the operating system, so that the operating system rides along in that channel to this incredible volume of customers. And then you take the fact they have made changes to the Java language . . . . So, if you want to target the Microsoft VM, which tags along on this incredibly powerful distribution channel, you pretty much have to use Microsoft’s tools.” Gosling, 12/10/98pm, at 44:19 - 45:18; see also id. at 43:20 - 44:9.

ii. As Gosling testified: "I certainly had developers talk to me who said that they had been developing with Visual J++ and then were surprised to discover that the software that they had developed was not portable. And besides this, the issue wasn't so much . . . the code that they themselves directly wrote. But a lot of it had to do with what libraries did they use, because if they used a library that was tainted with Microsoft-specific extensions, then their applications wouldn't run. And while the developer might have been aware when they wrote their own code whether or not they were using these things--they weren't necessarily aware, because a lot of the stuff was automatically generated by the tool, but whenever they incorporated libraries from other places, you know, they might have been sort of acquiring this problem, and it wasn't at all obvious when this problem would show up. And one of the unfortunate aspects of these extensions is that they are kind of an all-or-nothing thing. Namely, if there is just one place in the application that has got one of these Microsoft extensions, then that totally blocks portability." Gosling, 12/10/98am, at 58:15 - 59:11; see also Gosling, 12/10/98pm, at 39:1-16 (“There were, in general, no warnings that said, 'Oh, you're about to be nonportable.' There were some knobs that you could twist that were buried fairly deeply. And so, the natural thing . . . would be if you use their tools, boom, you are locked into their virtual machine.").

iii. An internal Boeing Corporation presentation in March 1998 described the likelihood of confusion as follows: “Microsoft continues to create Windows-specific Java implementations. Windows Foundation Classes (WFC) push the Microsoft Java Virtual Machine (JVM) into a 32-bit Windows-only environment. Developers using Microsoft programming tools may have difficulty knowing when they are creating Microsoft-JVM-only compatible Java programs.” GX 635, at TBC 000428.

iv. In a November 1996 e-mail discussion with Ben Slivka and others, Microsoft's Thomas Reardon suggested that promoting confusion was intentional: "As i told charlesf on the phone, at this point its not good to create MORE noise around our win32 java classes, instead we should just quietly grow j++ share and assume that people will take more advantage of our classes without ever realizing they are building win32-only java apps." GX 1332.

v. Similarly, a January 1997 article in CommunicationsWeek included in internal Microsoft e-mails to top executives reported the comment of a senior systems engineer at GTE Data Services as follows: "Management's fear, according to Hagerty, is the enticing power of the Microsoft environment. 'When developers get close to a deadline, there may be an irresistable temptation to take shortcuts . . . that will end up in the appliaction becoming Windows-specific.'" GX 519, at MSS 0080019.

c. Microsoft induced third parties not to support cross-platform Java

332. Microsoft used a number of strategems to induce third parties not to support, or to limit support for, cross-platform Java.

(1) Microsoft pressured Intel not to support cross-platform Java

333. Microsoft used a variety of means, including both threats to withhold Windows operating system support from Intel's microprocessors and offers to include Intel technology in the Windows operating system, to pressure Intel not to support cross-platform Java development, even when that development would have improved the performance of Java programs on Windows.

i. McGeady testified that Microsoft pressured Intel not to support Java "repeatedly and on multiple occasions." McGeady, 11/10/98am, at 14:19-21.

ii. Intel's Russell Barck also testified that "Microsoft believed JAVA was the enemy, from their perspective, and they felt we should consider it in the same light." Barck Dep., 8/25/98, at 45:16 - 46:8 (DX 2556); <u>see</u> <u>also</u> Barck Dep., 8/25/98, at 47:12-24 (DX 2556) ("Q: Were there statements made at those meetings at which you were in attend, that would give you an indication why Microsoft thought that Intel should not support JAVA? . . . A: I believe the primary reason was JAVA ran on multiple CPUs, or could run on multiple CPUs, and . . . it was an open-architecture platform. Q: If a program could run on multiple CPUs, is that something that would be detrimental to Intel? A: Not necessarily.").

iii. In March 1997, Jim Allchin, responding to an e-mail from Gates, wrote "I am positive that we must do a direct attack on SUN (and probably Oracle). I think we have to do this because we are just letting them grow their war chest. Between ourselves and our partners, we can significantly hurt their (certainly SUN's) revenue base. Yet, we are very silent. We are not getting the visibility that is required. I have talked to Compaq and HP about SUN and I know we can leverage them much more than we are today. We need to get Intel to help us. Today, they are not." Allchin repeated this statement at the end of the e-mail: "Finally, as I mentioned above, I think we HAVE to attack SUN (and Oracle). We need to do this at a company level. I just saw a billboard! from SUN in downtown Seattle over the weekend. SUN has a huge presence that we must stop." GX 356.

333.1. At a meeting between top Microsoft and Intel executives on August 2, 1995, before Microsoft had settled on its Java strategy, Bill Gates threatened to withhold Microsoft's support for Intel's next generations of microprocessors if Intel did not cooperate with Microsoft on platform and Internet issues.

i. Gates made clear at the August 2, 1995 meeting with Intel that "Microsoft would not support" Intel's "next processor offerings if we did not get alignment between Intel and Microsoft on platform issues" and communications issues, i.e., Internet issues. McGeady, 11/9/98pm, at 14:14 - 15:4. Intel's Ronald Whittier's minutes of the meeting similarly report: "Gates would not agree to let processors/OS's programs to progress unencumbered by platform/communications program issues . . . MS: Very sensitive to what Intel might do on the client side. Example: JAVA, a show stopper." GX 279, at MS CID 00077-78.

ii. McGeady, described by Whittier as the executive responsible for Java efforts at Intel in 1995, testified that Gates made clear that if Intel "started supporting Java, that would be a, quote, show stopper in the relationship. That would be another one of these things that would cause the show to stop, meaning the cooperation, presumably the cooperation, between Intel and Microsoft that we needed" in "support of our microprocessors, both the MMX microprocessor extensions as well as the P7 and other microprocessors." McGeady, 11/9/98pm, at 67:14 - 68:7.

iii. See also Maritz, 1/27/99am, at 27:22 - 29:6 (agreeing that Microsoft tried to stop Intel "from putting" its "resources into supporting work by either Netscape or Sun" and that Java was a "show stopper").

333.2. In the fall of 1995, after Microsoft had settled on the strategy to utilize Java as a Microsoft-specific technology, Microsoft enlisted Intel's help in preventing the "Java component model" from being established and convinced Intel not to disclose its Java work publicly at that time.

i. In a meeting on November 9, 1995, Paul Maritz told Intel executives, including McGeady, that Intel and Microsoft should cooperate to keep a "Java component model" from being established. GX 564, at 477MSCID00276 (McGeady notes); GX 285 (Landsman's notes) (Microsoft's concern is not the "Java language, per se," which Microsoft will support as "just another language" but the "Java component object model.").

ii. McGeady explained that Maritz's request was "a part of the embrace-extend model, which was, I believe, by this time they felt that it was unlikely that they could keep Java from happening at all, but they wanted to have it happen in a way that was incompatible. They wanted Microsoft's Windows Java to be incompatible with Sun Java. That would essentially have defeated the purpose--defeated the fact that someone could write a Java application and have it run on a cross-platform basis. If Sun's Java had one model and Microsoft's had a different model, then Java applications wouldn't work everywhere, and they wanted that to -- they wanted to keep" Java applications from "running anywhere in a compatible way. They wanted them -- they wanted Java applications written for Windows to just run on Windows, and Java applications conversely written for non-Windows platforms to not run on Windows." McGeady, 11/9/98pm, at 69:22 - 72:18.

iii. Frank Gill, McGeady's superior at that time in 1995, recorded Microsoft's request, as well as Intel's response -- an agreement to "not publicly endorse Java" -- in an e-mail written the day of the November 9, 1995 meeting: "Re: Java, we told them that we felt Java is on the way to becoming an Internet standard and felt a need to optimize around" Intel architecture "and particularly optimize our media components for the Java environment. They see this as supporting their mortal enemy . . . and argue SUN is our enemy as well. Paul describe to me as 'this is equivalent to us optimizing for powerpc'" a competitive chip. I "left it with him that have not starting anything until we talked today, we will not publicly endorse Java, and I needed internal review before authorizing this work to commence. So, bottom line, this is BIG DEAL to them." GX 284.

iv. McGeady, 11/12/98am, at 69:14-17 ("Well, yes, actually. We had hoped to support Java in a more public fashion . . . . MS was able to convince Intel's management to withhold its public support for the program."); see also McGeady, 11/9/98pm, at 73:2 - 74:22 (Microsoft succeeded in having Intel withhold public support for Java).

333.3. Throughout 1996 and 1997, Microsoft continued to push Intel into supporting its Microsoft-specific implementation of Java and away from compatibility with the Sun model.

i. In April 1996, Intel's Craig Kinnie reported that John Ludwig of Microsoft "also wanted to convince me that our media class library work was aiding the competition to the Win/IA platform and they considered our work as competitive." GX 567.

ii. McGeady explained this comment: "Q: Did you have an understanding in April of 1996 as to what that was referring to? A. Yes. They were very upset that we were taking our optimized audio, video, 3-d graphics software and adapting it to work within Sun's Java framework and environment, and they wanted us to stop. They considered it competition." McGeady, 11/10/98am, at 13:18-23; see also McGeady, 11/10/98, at 8:3-20 (Microsoft strongly pushing toward the Microsoft model and "away from our stated architecture and strategy, which was to adhere to the Java standards and be compatible with Sun").

333.4. Microsoft also pressured Intel not to assist Netscape with Intel's Java virtual machine, even though Netscape Navigator was still a leading Windows application and Intel's work would have improved the performance of Navigator on the Windows operating system.

i. In an April 1996 memorandum, Intel's Alan Holzman summarized requests made by Microsoft's Bob Muglia and John Ludwig: "Muglia/Ludwig want exclusive access to Intel Architecture's VM work: Ludwig does not want us to give Netscape our Java virtual machine work. Netscape access to VM is VERY touchy with MS!!!" GX 566.

ii. McGeady explained that Holzman's comments reflected that "Microsoft was very upset that we were working on Java at all, and in particular, they didn't want -- we had a very highly optimized virtual machine that ran very fast on Intel architecture, and they did not want that VM to become part of Netscape Navigator." McGeady, 11/10/98am, at 6:9-16.

333.5. In June 1996, Bill Gates convinced Intel not to "wrap" Windows APIs as Java APIs, i.e., not write Java APIs that would work well with Windows, even though doing so would improve the performance of Java programs on Windows.

i. Eric Engstrom admitted, with regard to Microsoft's efforts (GX 235) to dissuade Intel from "helping Sun create Java multimedia APIs," that Microsoft's conduct had an impact on Intel's development work: "Q: Well, in fact, you were successful for some period of time in convincing Intel to stop helping Sun create Java multimedia APIs, were you not? A: For some period of time we succeeded in working roughly along the lines of this e-mail, yes." Engstrom, 2/23/99pm, at 33:19-23.

ii. In a meeting in June 1996, Gates told Intel CEO Andy Grove that Intel should not "take anything resembling a Windows API and wrap it as a Java API." GX 289 (6/6/96 Gates e-mail to Maritz). Grove acquiesced. GX 289.

iii. As McGeady testified, the sole purpose for not wrapping Windows APIs as Java APIs was to harm the prospects of cross-platform Java. He explained that "the bottom line was we were -- we felt we were taking our multimedia technology, interfacing with the underlying basic capabilities of windows and providing that multimedia technology as a integral part of Sun's Java. That, I interpret, is what Bill was complaining about." McGeady, 11/10/98am, at 15:21 - 16:12.

iv. McGeady further testified that wrapping Windows APIs as Java APIs would benefit users and developers because: "You would have a Java implementation that was both compatible with and would be able to run the same applications on a macintosh and on a windows machine, and you would have a java implementation that would run faster on a windows machine than one that was implemented in a way that wasn't cognizant of the underlying Windows APIs. So it would be faster and it would be more compatible." McGeady, 11/10/98am, at 15:21 - 17:3.

333.6. Microsoft continued its campaign to pressure Intel not to support cross-platform Java in 1997.

333.6.1. In February 1997, Bill Gates suggested to Intel CEO Andy Grove that, if Intel wanted Microsoft not to support a competitive chip, Intel would have to stop assisting Sun with Java multimedia.

i. In February 1997, Microsoft was asked by AMD, an Intel competitor, to support an AMD technology called "3DX." When asked by Jim Allchin if Microsoft should support the technology although Intel would not be in favor of such support, Gates responded that, "If Intel has a real problem with us supporting this then they will have to stop supporting Java Multimedia the way they are. I would gladly give up supporting this if they would back off from their work on JAVA which is terrible for Intel. I have a call with Andy on this topic coming up on Monday." GX 290.

ii. In February 1997, Gates wrote to Paul Maritz about "Intel and Java." "I called Andy today to talk to him about our sensitivities about JAVA and asking what goal Intel has in working on JAVA. I told him we thought they were hurting themselves but [sic] doing any development other than just pure tuning things for Intel." GX 935.

333.6.2. In the spring of 1997, Microsoft convinced Intel to stop helping Sun develop Java multimedia APIs, "especially those that run well on Windows."

i. In a May 1997 status update on Microsoft plans with Intel, Microsoft's Eric Engstrom identified a primary Microsoft objective for Intel as: "Intel to stop helping Sun create Java Multimedia APIs, especially ones that run well (ie native implementations) on Windows." GX 235. To achieve this goal, Microsoft proposed to Intel that it would include Intel technology in Windows in return for Intel's agreement to stopping helping Javasoft write multimedia APIs. GX 235 ("Intel agrees to stop helping Javasoft in any area where Microsoft agrees to ship Intel technology as part of DirectX media," and "augment DirectX rather than compete with it . . . . We want them pushing DirectX rather than some bits of Javasoft and Intel technology.").

ii. Engstrom testified about his May 1997 meeting with Intel concerning Java multimedia APIs as follows: "Q: Let me be sure I understand your testimony. Your goal was to try to convince Intel to work with Microsoft's multimedia and not with Sun's competing Java multimedia; correct, sir? A: My goal was to get them to work with us. You know, I wanted to be the best partner I could be for Intel. Q: And, in fact, sir, you wanted more than that, didn't you? Didn't you also want Intel to stop helping Sun create Java multimedia APIs? A: Yes, that would be a goal of being a great partner for them, is that, you know, they have no bandwidth for other partners. . . . Q: Let me just be sure I understand your testimony before we move on. In this particular case, it was Microsoft's goal absolutely to stop Intel from working with the competitor, someone who created competing APIs to Windows, even if those APIs ran well on Windows; is that what you said? A: It was my goal to get Sun -- to get Intel to work with Microsoft for their multimedia investments, not with Sun. Absolutely." Engstrom, 2/23/99pm, at 21:14 - 22:10, 35:2-23.

iii. Although Engstrom denied that he reached agreement with Intel, he admitted that Microsoft was successful in stopping Intel from helping Javasoft with multimedia Java APIs for some period of time. Engstrom, 2/23/99pm, at 33:11-23.

333A. Contrary to Microsoft's unsubstantiated assertion (MPF ¶ 941), Microsoft's repeated efforts to dissuade Intel from supporting cross-platform Java had a significant adverse impact on Intel's Java support and development work.

i. GX 284 (11/9/95 Frank Gill (Intel) e-mail; in reponse to Maritz's statements that any Intel support of Sun was supporting Microsoft's "mortal enemy" and "the equivalent to [Microsoft] optimizing for powerpc," Gill told Maritz that Intel would not publicly endorse Java and Gill would seek "internal review" before authorizing the Java work to commence).

ii. McGeady, 11/12/98am, at 69:14-17 ("Well, yes, actually. We had hoped to support Java in a more public fashion . . . . MS was able to convince Intel's management to withhold its public support for the program."); <u>see</u> <u>also</u> McGeady, 11/12/98am, at 62:22 - 63:5 (Microsoft succeeded in having Intel

withhold public support for Java).

iii. Andy Grove, the CEO of Intel, acceded to Gates's 1996 pressure on Intel not to "wrap" Windows APIs as Java APIs -- pressure specifically intended to harm cross-platform Java and to impair its performance on Windows. GX 289.

iv. Engstrom admitted with regard to Microsoft's efforts (see GX 235) to dissuade Intel from "helping Sun create Java multimedia APIs," Microsoft's conduct had an impact on Intel's development work: "Q: Well, in fact, you were successful for some period of time in convincing Intel to stop helping Sun create Java multimedia APIs, were you not? A: For some period of time we succeeded in working roughly along the lines of this e-mail, yes." Engstrom, 2/23/99pm, at 33:19-23.

v. McGeady summarized the three years of Microsoft pressure as follows: "These meetings that we had had with Microsoft concerning Java and the Internet represented a level of what I considered to be predatory competition beyond what I had ever seen." McGeady, 11/12/98am, at 82:17-20.

334. Microsoft's efforts to prevent Intel from supporting cross-platform Java made no business sense except to preserve Microsoft's monopoly power and significantly impeded Intel's support and development efforts for cross-platform Java.

i. Bob Muglia conceded that it is not in Microsoft's interests to have developers write to "Sun's Java platform," even if the resulting programs work well on Windows. Muglia, 2/26/99pm, at 79:8 - 80:15.

ii. As Steven McGeady put it, based on his meetings with Microsoft executives, "they wanted Java applications written for Windows to just run on Windows, and Java applications conversely written for non-Windows platforms to not run on Windows." McGeady, 11/9/98pm, at 69:22 - 72:18.

iii. As Professor Franklin Fisher summarized, Microsoft sacrificed its interest in improving Windows in pressuring Intel not to support cross-platform Java efforts: "The more things that will run well on Windows, presumably, the better is Windows functionality, . . . the more attractive Windows will be to users. Here is something that suggests that Microsoft was particularly not interested in having something that would run well on Windows." Fisher, 1/7/99pm, at 52:19-24.

335. In light of Mr. Gates' contemporaneous writings and the other evidence, his denials during his deposition of any effort to keep Intel from supporting Java are incredible.

i. Compare (Gates Dep. played 11/9/98am), at 62:21- 64:23 ("Q. Did Microsoft make any effort to convince Intel not to help Sun and Java. A. Not that I know of."), with GXs 279, 289, 290.

(2) Microsoft pressured Apple and IBM not to support cross-platform Java

336. Microsoft sought to use its leverage as Apple's primary application supplier, and the leverage that stems from the need for QuickTime to be technically compatible with Windows, to secure Apple's support of Microsoft's Java strategy.

i. Microsoft linked its cooperation with respect to continuing the development of Microsoft Office for the Macintosh operating system to Apple reducing its support of cross-platform Java. As Microsoft executive Don Bradford wrote on January 21, 1998 to Bill Gates and others: "Mac Office is the biggest Apple carrot. From a browser, mail client and Java perspective, Apple has other options and will try to sit on the fence. Negotiating from a single point, centered around Mac Office, will give us better leverage." GX 267.

ii. As Bradford wrote on February 13, 1998 to Gates and Maritz, as well as other Microsoft executives, with respect to Microsoft's August 1997 agreement with Apple: "We get our APIs deployed in perhaps the only VM on Mac, reinforcing Java-is-a-language-not-a-platform, while minimizing our investment." GX 268.

iii. On April 28, 1998, Gates wrote his executives, and again emphasized that Microsoft should use its cooperation with Apple on its QuickTime software as an inducement to get Apple to support Microsoft's Java strategy: "I would love to see enough convergence of QT and our equivalent so that . . . we use this 'convergence' to get more JAVA/IE affinity from Apple." GX 270.

337. Similarly, Microsoft sought to use its operating system monopoly to ensure that IBM would not support cross-platform Java development.

i. In October 1997, Gates wrote to Joachim Kempin "that the Java religion coming out of

> the software group is a big problem . . . . Overall, we will never have the same relationship with IBM that we have with Compaq, Dell, and even HP because of their software ambitions. I could deal with this just fine if they weren't such rabid Java backers. We are told they have funded Netscape to do the browser without e-mail and they are proposing to fund Netscape to give their browser away. They have also funded a lot of SUN stuff . . . . I am willing to take some risk in improving the relationship and think you should approach them on steps for improvement. You should not position it as 'overnight we will treat you like Compaq.' We should position it as let's do some things that are good for both of us but which require some of the rhetoric to be lowered on both sides. On their side I mean Java and NC." GX 257.

ii. As Garry Norris of IBM testified, that Bill Gates' statements were "very consistent" with the type of treatment IBM received from Microsoft in the 1996 time frame, specifically, pressure not to offer software products that competed with Microsoft's products. Norris, 6/8/99am, at 59:4 - 60:4.

iii. <u>See</u> <u>also</u> (Gates Dep. played 11/17/98pm), at 44:15 - 46:25 (Q: Mr. Gates, let me put it this way. In October of 1997, were you trying to get IBM to reduce its public support for Java? A: I say in here that under some circumstance the rhetoric should be lowered on both sides and that I think that's -- you know, that makes sense in certain circumstances . . . . Q: And by "rhetoric," you're talking about public rhetoric, were you not? A: Definitely public rhetoric.").

338. Microsoft's pressure on Apple and IBM is unjustified by any legitimate business interest.

338.1. The contemporaneous evidence demonstrates that Microsoft's goal was to undermine cross-platform Java, not to enhance Windows (except by defeating the cross-platform threat).

i. On August 8, 1997, Apple's Avadis Tevanian sent an e-mail to Microsoft referencing an earlier Microsoft/Apple agreement, "Java collaboration," and requesting that Microsoft stop "trying to lock-out QuickTime from Windows." Gates forwarded the message to Paul Maritz and other Microsoft executives with the following comment: "I want to get as a much mileage as possible out of our browser and JAVA relationship here. In other words a real advantage against SUN and Netscape. Who should Avie be working with? Do we have a clear plan on what we want Apple to do to undermine Sun?" GX 265.

ii. In an August 21, 1997 e-mail reporting on "conversations with billg last nite," Microsoft's John Ludwig made clear what Microsoft's plan was: "On the Java work, Bill says the ball is in our court to come up with a win/win solution. He suspects that we should be cooperating only at the lowest level -- native code interfaces, jits, etc., not at the class lib level. Bill was clear that his whole goal here is to keep Apple and Sun split. He doesn't care that much about being aligned with Apple, he just wants them split from other potential allies." GX 255.

iii. As Gates put it in January 1998 : "There is a big question of what we should do with JAVA on Mac — whether doing more work and working closely with Apple could help us in this 'battle'. I think it can and would hate for Apple to have to go back to the SUN camp." GX 897.

338.2. In light of the contemporaneous documentation of Microsoft's efforts to "keep Apple and Sun split," Gates' denial in his deposition ((Gates Dep. played 11/2/98pm), at 14:14 - 17:6; 18:18 - 24) of any involvement in any plan by Microsoft to encourage Apple to undermine Sun's Java is not credible.

(3) Microsoft entered into agreements with ISVs that limited their ability to support cross-platform Java

. 339. In addition to its First Wave agreements, Microsoft entered into other agreements to restrict distribution of non-Microsoft JVMs and native code interfaces by ISVs.

339.1. Microsoft entered into agreements with third parties to support exclusively Microsoft's JVM and native code interfaces. Those agreements were enjoined in November 1998.

i. For example, an unsigned License Agreement between MS and Fujitsu for redistribution of MS Java software development kit includes the following obligations on Fujitsu: "3. Obligations. During the term of the Agreement, Company shall: (a) redistribute the Microsoft virtual machine for Java included in the Licensed Software (the 'MS Java VM') as part of Company Product and not any other virtaul machine; (b) use only the Microsoft native code interfaces (J/Direct, RNI, Java/COM) that are part of the MS Java VM for any native code calling; (c) support Internet Information Server for server-side functionality; (d) make reasonable efforts, where appropriate, to redistrbution the Microsoft Visual J++ trial edition with Company products." GX 1331, at MSS 0165897-98; see also Muglia, 2/26/99pm, at 59:19 - 62:12 (Microsoft and AimTech agreement).

ii. Sun Microsystems, Inc. v. Microsoft Corp., 21 F.Supp. 2d 1109, 1127-28 (N.D. Cal. 1998) ("Since the court finds that Sun is likely to prevail on the merits and that it may suffer irreparable harm if Microsoft is not enjoined, a preliminary injunction is hereby issued against Microsoft, and its officers, agents, servants, employees, attorneys, and those in active concert or participation with them who receive actual notice of this order by personal service or otherwise, pending trial, from: . . . (E) Conditioning any license to any Microsoft product on exclusive use or distribution of Microsoft's Java virtual machine; (F) Entering into any agreement, condition or arrangement with any third party that requires such third party to exclusively use Microsoft's interfaces to its runtime interpreter when invoking native code.").

339.2. Microsoft also restricted RealNetworks, the leading vendor of Internet streaming software, from supporting Netscape and Java technology.

i. The July 18, 1997, agreement between Microsoft and RealNetworks requires, among other things,

GX 1369, at MS8 000635, MS8 000637 (sealed).

ii. RealNetworks' Jacobsen explained: "In essence, Microsoft has rights of first discussion if we're going to go support something that would compete with DirectDraw or DirectShow. The impact on us is to put a speed bump in discussions with Sun or Netscape because before we could conclude an agreement with Sun or Netscape, we would have to go to Microsoft to have discussions with them." Jacobsen Dep., 1/13/99, at 148:6-14. Microsoft, he further explained, would be given the opportunity to propose an alternate to the Sun or Netscape interface, and only if that solution did not work would RealNetworks be permitted to support Sun or Netscape interfaces. Jacobsen Dep., 1/13/99, at 149:21 - 151:4. If this clause were invoked, Jacobsen testified, he expected that Microsoft "would engage is some serious discussions in between carrots and sticks" to convince RealNetworks not to support the competitive interfaces. Jacobsen Dep., 1/13/99, at 150:23 - 151:4.

4. Microsoft's efforts to impede cross-platform Java facilitated the maintenance of Microsoft's operating system monopoly, hindered innovation, and harmed consumers

340. Microsoft's conduct reduced the likelihood that cross-platform Java could erode the applications barrier to entry and thus facilitated maintenance of Microsoft's operating system monopoly.

i. John Soyring testified that Microsoft's Java-related conduct has had "a retarding effect on the acceptance and the growth rate of Java in that our clients, at least at a minimum, pause to think, is Java acceptable since it may not be able to achieve this objective of `write once run anywhere.' A second aspect is from a commercial software developer program, commercial software vendors who want to develop and sell an application that carries the logo or the mark design for Microsoft Windows 95 and Windows NT. In that contract, I understand, is the provision that requires that the vendor, if they use Java as a language, write it to the Microsoft Java and ship the Microsoft Java virtual machine with their application. That poses, then, a fear amongst our customers that Java application may only run on these Microsoft platforms and, thus, not be cross-platform capable as advertised by the other Java components." Soyring, 11/18/9pm, at 53:13 - 54:7.

ii. David Clare, an executive with Novell Corporation, explained in his deposition that the fragmentation of Java is inherently harmful to customers: "So Java, again, provides a unique opportunity -- where everyone's trying to sell platforms, right, we sell NetWare, they sell Solaris, they sell -- but we are trying to ensure that there is some consistency, you know, so that we can ultimately help our customers because it's not going to help the customers if we fragment Java, right? . . . . Novell is very committed to -- to maintain and preserve the Java as the consistent implementation across different platforms." Clare Dep., 10/29/98, at 63:12 - 63:20, 79:16 - 81:3(DX 2613).

iii. In October 1997, Ken Morris, Chief Technology Officer of PeopleSoft, a significant Java application developer wrote to Steve Ballmer and Alan Baratz (President of JavaSoft): "We are concerned that the current course of events may result in competing definitions of Java. This splintering of Java would destroy one of the most powerful benefits of the technology -- that it is based on a universally agreed upon standard. Competing Java definitions would force every application developer, including PeopleSoft, our competitors, other software component vendors, and in-house systems designers, to divert valuable development resources towards porting efforts at the expense of product innovation, much like the situation today with the various implementations of Unix." GX 1324.

iv. James Gosling testified: "One of the goals of the whole Java technology is to save developers the cost of developing on every possible virtual machine. One of the clear goals of the Java compatibility tests is to insure uniformity amongst all of these virtual machines so that when a developer develops a program, they don't have to test on this virtual machine and that virtual machine and the other virtual machine. The situation that Microsoft's actions put us in is one where developers, by and large, can end up having to test twice: once on a certified VM to see that they will work across, you know, Solaris, OS/2, Apple, or whatever, and then they separately have to test for Microsoft because of the choices that Microsoft has made." Gosling, 12/10/98am, at 23:20-24:13.

v. Netscape's Richard Schell testified that it is "of great concern that Microsoft had gone down its own path. Ideally, in an ideal world, we would have cooperated and collaborated with Microsoft to ensure that we all provided a -- a common implementation. It was not our desire that there be multiple implementations. It clearly was, to me, clearly was Microsoft's intention to provide at least two, and that made it sort of difficult, but at least we would have one that was cross-platform and hopefully could influence Microsoft to -- to comply and conform with that cross-platform implementation." Schell Dep., 9/15/98, at 71:15 - 72:9 (DX 2587).

vi. Gosling summarized the importance of browser competition and unfragmented Java to software developers and users: "The preservation of a competitive market for Internet browsers is critically important to preserving the viability of the cross-platform Java technology. Microsoft's actions injure not only Sun, but the entire community of Java-based developers, computer users, and potential operating system competitors of Microsoft . . . . If Microsoft's broad distribution of its Microsoft-dependent implementation of the Java technology in Windows, in Internet Explorer, and in its development toolkits leads Java developers to use these Microsoft dependencies, the potential of this technology to reduce the barriers to developing new operating systems will be undermined, and Microsoft's operating system monopoly will be further enhanced. In short, if Microsoft successfully fragments the Java technology, the cross-platform benefits to vendors, developers, and users of the Java technology will be damaged, and any threat that the Java technology poses to Microsoft's dominant Windows operating system will be neutralized." Gosling Dir. ¶¶ 65, 73-74.

vii. Soyring also testified that Microsoft's fragmentation of Java reduces the possibility that cross-platform Java will be able to serve as a foundation for renewed operating system competition: "Microsoft licensed Java technology from Sun, and has released its own Java implementation. Microsoft tuned its version of Java for Windows, inhibiting the potential for application developers to write applications once and have them run on many different operating systems. Wide use of Microsoft's version of Java negates the potential that Java might undermine Windows application advantage." Soyring Dir. ¶ 28.

viii. Professor Fisher summarized the impact of Microsoft's conduct: "Now, they didn't eliminate for all time any possible threat, but they took actions in both the browser area and the Java area to insure that what at least they saw as a threat didn't materialize very fast. . . . It's also the case that their actions toward Java have been such that, at least for the time being, Java, combined with browsers or not, doesn't seem very likely to become an alternative platform. It's been substantially impeded." Fisher, 1/6/99am, at 35:13 - 36:1; see also Fisher, 1/12/99pm, at 22:4-9 ("Microsoft has, I think, also impeded the development and spread of Java. Whether that has successfully blunted the Java threat as opposed to merely slowed it down, I don't think one can say at this time, but I think it certainly has done a good deal in that direction.").

341. Microsoft's argument that its conduct with regard to Java has had little competitive impact, because the prospect of cross-platform Java is illusory cannot be reconciled with the evidence.

341.1. Microsoft's suggestion that Sun's objective is not to encourage competition in operating systems -- but rather to eliminate operating system competition altogether and favor its own operating system or "put Java directly on the chip" (Muglia Dir. ¶ 32; Muglia, 2/26/99pm, at 10:5 - 11:22; Martiz Dir. ¶ 244) -- is inconsistent with the evidence and not relevant to assessing the competitive implications of Microsoft's conduct.

i. Gordon Eubanks, former CEO of Symantec, a significant Java tools and runtime vendor, emphasized the cross-platform, not Sun-specific, potential of Java: "One of the great things with Java is that when you create a Java application, it will run on any machine that has a Java virtual machine. That was one of Sun's contributions. Bill Joy was really the spirit behind this, one of the great scientists at Sun. And he is right about this. If you create an applet, it can run on any machine. And, of course, that is perfect for the Web because you don't know when you're downloading something, what machine it's going to be running on." Eubanks, 6/16/99am, at 68:11-20.

ii. Maritz sought to portray Java as a threat specifically in concert with Sun's JavaOS, but could do so only by describing Java as a cross-platform threat, with JavaOS as merely one example of an alternative operating system whose success could be facilitated by the success of Java: "And the the second point is the point that I spoke about earlier with respect to middleware, that one of the ways that threatens Windows is it provides a means to move the applications off the Windows platform onto another operating system. And, in particular, Sun Microsystems is developing an alternative operating system that I have labeled there the 'Java OS.' And I believe, in fact, there is some cooperation between Sun and IBM to develop that Java OS operating system. "Q: And what exactly enables the replacement of Windows by Java OS, as you suggest by your diagram? A: The point here is that you can develop a version of that Java runtime that will run on the other operating systems. You have one version of it that runs on Windows, and another version that runs on the other operating system, or another operating system. But the applications, in either case, just see the middleware layer. That makes them easy to move to the other operating system, thereby enabling the other operating system to compete with Windows." Maritz, 1/28/99am, at 60:9 - 61:2.

iii. Gosling also made clear that Sun's goal was to ensure that Java implementation "could run on top of many other platforms: on top of the windows platform, on top of the mac platform, and so forth." Gosling, 12/2/98pm, at 47:3-9; see also Gosling 12/2/98pm, at 67:21-25 ("the way that you phrase that question was as though our goal was to take over one hundred percent of the entire universe. And actually, I think we would have been happy with two orders of magnitude less than that."); Gosling, 12/2/98pm, at 68:15-21 ("One would never actually do something like that" put Java directly on the chip, "I mean, the Java virtual machine was designed as something . . . that's easy to adapt to many kinds of hardware. It's actually a rather poor design for any specific piece of hardware, so one would never actually get rid of the virtual machine completely."); Gosling, 12/2/98pm, at 77:5-7 ("taking the Java virtual machine and doing a pure silicon implementation of it would be, actually, a nutty thing to do, to use a technical term."); Gosling, 12/2/98pm, at 78:17-18 ("It turns out that the more of the virtual machine you put into silicon the slower it goes . . . what they were trying . . . to do is not achieve high speed but small size.").

iv. Indeed, as Dr. Gosling explained, Java could stimulate competition among CPU chips, not by displacing operating systems, but by freeing microprocessor vendors of limitations inherent in designing for particular operating systems: "I mean, one of the things you have to understand about the Java architecture is that it makes applications that are distributed by developers independent of the underlying platform. Independent of not only of the operating system, but of the specific chip that's there." Gosling, 12/2/98pm, at 61:9-63:14.

341.2. Microsoft's suggestion that cross-platform Java is flawed (Muglia Dir. ¶¶ 10, 17-

18) is also inconsistent with the evidence and not relevant to assessing the competitive implications of Microsoft’s conduct.

341.2.1. The evidence, including evidence from Microsoft, shows that cross-platform Java presented a significant platform threat.

i. See supra Part III.C.; ¶¶ 57-59.

341.2.2. That cross-platform Java cannot yet fully replicate the applications available for Windows means simply that the technology is evolving and reflects that Microsoft has succeeded in retarding its development.

i. Gosling testified that: “statistically speaking, in both of these cases of source compatibility and binary compatibility, the number of problems--if you survey a large number of applications, the number of problems is statistically very small." Gosling, 12/3/98am, at 49:18-23; see also Gosling, 12/3/98am, at 32:14-18 ("While we are not in a state of being perfect in achieving that goal, we are significantly better than we were in the past, and we are continuing to get better.” Gosling also explained that Java has improved significantly over time: "As people's implementations of Java runtimes have gotten better and better, you know, . . . the need to do native methods purely for performance has become much, much less frequent than it was . . . . There exists a small number of cases, although with the latest generations of things, actually going to native methods can actually slow you down." Gosling, 12/2/98pm, at 6:3-15; see also Gosling, 12/2/98pm, at 7 (noting that DX 2045, an article complaining about Java performance, reflects the state of affairs several years ago); Gosling, 12/2/98pm, at 12:3-14 (noting, with respect to DX 1926, an article stating that Java has poor performance, particularly on Win16: "I've spent a lot of time with corporate customers around the world. There are hundreds of thousands of people developing Java programs, and doing so quite happily and quite successfully. And, you know, for a lot of people, the actual raw performance of Java is not often the number one concern. You know, it's just is when you and buy a car. You don't just go and buy the fastest car. You may buy the car that is the most reliable, or you might have--buy the car that's got the most comfortable seats. Yes, there have been performance issues. They're getting better.").

ii. Dean Richard Schmalensee acknowledged that "many companies are aggressively pursuing the promise of Java." Schmalensee Dir. at ¶ 144.

iii. Executives at Symantec also recognized the viability of cross-platform Java, particularly compared to the difficulties of writing platform-specific code, notwithstanding the immaturity of Java: "Admittedly we're now at the stage where it's 'write once, debug everywhere.' However, that's still far more productive than 'write once, re-write everywhere' for the C++ and VB/VJ++ developers." GX 2078 (March 1998 Symantec e-mails including draft of public statement ultimately not released because, although "confirmed that everything . . . in this statement is true and can be proved," "it's better to say nothing than risk the blast from MS.").

341.2.3. Indeed, the contemporaneous documents do not corroborate Microsoft's argument that Java specifically, and cross-platform programming generally, is inherently flawed in any way (e.g., "least common denominator" limitation (MPF ¶ 864), slow performance (MPF ¶ 865), or incompatible implementations (MPF ¶¶ 866-867)). To the contrary, they indicate that cross-platform Java can provide performance and functionality rivaling platform-specific programs.

i. As Microsoft's Peter Kukol reported to Gates and Muglia in October 1996: "Incidentally, we've been doing quite a bit of research on how to make Java run as fast as C++ across the

board -- this requires pretty advanced program analysis and optimization work but should be quite feasible (in fact, it might be possible to make Java run faster in many cases -- after all, it's a much cleaner language plus the code generator can easily work on much larger parts of the program)." GX 1341.

ii. Similarly, in a February 1997 overview of "results of VJ++/Java focus groups," Microsoft's Hyer Bercaw summarized developers' views of Java: "The productivity of the language is recognized, but is secondary to the cross-platform benefits. . . . Most people thought that performance degradation of Java code

from platform to platform was acceptable (and to be expected -- not everyone is running a Pentium Pro). . . .” GX 2518.

iii. James Clark, former Chairman of Netscape, explained why the mere speed of Java programs may not be of ultimate competitive significance: “Q: If you did try to build a truly cross-platform piece of software that had to cope with the differences in APIs exposed by different operating systems, do you agree with me that the functionality of that piece of software would by definition be limited? . . . A: Not necessarily. In fact, you could argue that the fact that it's available in multiple platforms makes it even less limited because then it could communicate. It's a communications product. It would be as though you had one telephone of a particular brand that wouldn't communicate with another telephone of a different brand.” Clark Dep., 7/22/98, at 118:4-17 (DX 2562).

iv. <u>See</u> <u>also</u> GX 1342 (JavaWorld article, March 98, suggesting Sun's new dynamic compiler will be much faster than current JIT compilers, and faster than everything but compiled and optimized C); GX 1343 (JavaWorld article entitled “Performance Tests Show Java as Fast as C++,” February 1998: "To our shock, we rarely found any differences in speed at all. Where Java is significantly slower than C++, it's due to Java's stringent security model or to garbage collection. . . . Therein lies Java’s bad reputation. Most performance perceptions for Java were derived from older JVMs . . . . The perceptions earned by earlier JVMs are no longer valid, since most JVMs are delivered with JIT compilers."); GX 1344 ("Java performance is good enough for the majority of today's web applications. In addition, by 2001, Java performance will cease to be a major barrier to adoption of the technology for better than 95% of deployment opportunities . . . "). <u>See</u> <u>generally</u> GX 1342-1356 and GX 1362-1363 (press articles praising newer versions of Java).

341.2.4. Microsoft’s suggestion that variations in non-Microsoft implementations of Java, including Netscape’s, indicate that cross-platform Java is illusory (<u>see</u> MPF ¶ 867) ignores the relevant evidence.

i. As Netscape's Richard Schell pointed out in his deposition: "Q: At the time you left the company in February of '98, did the Java virtual machine in Communicator, any version, implement the Java Native Interface? A: I believe it passed the Sun certification tests, except in a few obscure cases, yes. Q: Did you have any discussion with Mr. Baratz or anyone else at Sun about the consequences of Netscape's failure to comply strictly with each and every one of the Java certification requirements? A: Since I had been part of the contract negotiation, we were very familiar with exactly what the process was for going through certification and reaching compliance. We also were pragmatic enough to know that there were time periods in which we would be out of sync, and we worked to get back into sync any time that that happened. Q: Okay. But my question to you, Dr. Schell, was whether you talked to Mr. Baratz or anyone else at Sun about what would happen if you were not in full compliance with the Java certification requirements? A: We knew the consequences. We had always cooperated and got back into compliance." Schell Dep., 9/15/98, at 174:8 - 175:10 (DX 2587).

ii. An August 1997 memorandum from IBM's John Thompson, now CEO of Symantec Corporation, to Sun's Scott McNealy, Netscape's James Barksdale, and others, made clear the distinction between Netscape Navigator and Microsoft Internet Explorer: "We must maximize the distribution of 100% Pure Java run time environments by focusing on the following: Promoting Netscape Navigator as the only browser that is committed to supporting 100% Pure Java. . . ." DX 1894.

341.2.5. Microsoft's reliance on the failure of early efforts by Corel and Netscape to create Java versions of their application programs to suggest that cross-platform Java is inherently flawed (MPF ¶ 865) is misplaced. Indeed, Netscape abandoned its Java program because of Microsoft's predatory conduct, not flaws in Java.

i. With respect to Corel Corporation's decision not to continue development of a full-fledged Java productivity application suite

in 1997 (see DX 1930 (press article)), Gosling explained that "all that is saying is that Corel built a word processor that was not fully functional and that earlier on they had been touting one of its advantages being that it was written in Java. That is an advantage, but the fact they didn't follow through, that they didn't bother to build a fully functional word processor is, I think, what that is really saying. It's not actually a failure of Java. . . . the very next paragraph . . . says Corel says its retrenchment was not a negative statement of Java's capabilities." Gosling, 12/3/98pm, at 22:13-23; see also Gosling, 12/3/98pm, at 23:21 - 24:5 (noting that Corel's problems were before the high performance VMs arrived, and more related to Corel's design decisions than Java).

ii. Barksdale made clear that Microsoft’s predatory conduct, not technical problems with Java, was the primary reason for Netscape’s decision to curtail its Java software development effort: Netscape canceled the Java-based Navigator (i.e., “Javagator”) “when we had to cancel most of our -- some new plans for the browser as a result of downsizing our staff as a result of making the client free and cutting back on our expenditures.” Barksdale, 10/26/98am, at 32:4-7; see also id. at 35:2-5 (“Q: And isn't it true that that project was abandoned because of performance problems, notably the speed at which the product ran? A: No.”).

iii. Netscape’s Richard Schell also explained that writing in Java had not posed significant technical difficulties for Netscape. Schell Dep., 9/15/98, at 57:20 - 58:21 (DX 2587).

iv. Soyring described his understanding, based on IBM’s interest as a potential purchaser of a Java-based browser, of the viability of a Java-based browser and the reasons for Netscape’s abandonment of the Javagator project: “I can comment on Javagator. We were developing--or we do--are developing a product which does have a requirement for a Java-based browser--in other words, a browser that's been implemented in the Java language--and we are looking for sources to be able to potentially buy that technology or license that particular technology. Javagator was one of the alternative sources we were considering. And until Netscape ran into financial problems where part of their revenue stream dried up, they were going ahead and developing it. They stopped developing it soon

> **after they had to start giving away their browser, or at least they decided to give away their browser for a variety of reasons." Soyring, 11/18/98pm, at 25:7-20.**

341.3. Microsoft's suggestion, that AOL will partner with Sun to develop a Java-based version of Navigator, is inconsistent with, and itself refutes, its argument that Java is inherently flawed.

i. Muglia pointed out: "Several press reports indicate that Sun will rewrite Netscape Navigator entirely in Java . . . . Press reports have quoted Sun management as saying that part of the agreement will be the development by Sun of a 'pure Java' version of Netscape Navigator browser, which will then be distributed by AOL to all of its customers.". Muglia Dir. ¶¶ 20, 97.

ii. Similarly, Dean Schmalensee appeared to focus on AOL's potential distribution of Java virtual machines as of competitive significance: "It's my understanding that AOL has indicated it's going to distribute . . . a very large number of Java virtual machines . . . as part of its client software." Schmalensee, 6/21/99pm, at 95:16-23.