EXHIBIT B

Findings of Fact given collateral estoppel effect by Judge Motz:

2. An “operating system” is a software program that controls the allocation and use of computer resources (such as central processing unit time, main memory space, disk space, and input/output channels). The operating system also supports the functions of software programs, called “applications,” that perform specific user-oriented tasks. The operating system supports the functions of applications by exposing interfaces, called “application programming interfaces,” or “APIs.” These are synapses at which the developer of an application can connect to invoke pre-fabricated blocks of code in the operating system. These blocks of code in turn perform crucial tasks, such as displaying text on the computer screen. Because it supports applications while interacting more closely with the PC system's hardware, the operating system is said to serve as a “platform.”

4. An operating system designed to run on an Intel-compatible PC will not function on a non-Intel-compatible PC, nor will an operating system designed for a non-Intel-compatible PC function on an Intel-compatible one. Similarly, an application that relies on APIs specific to one operating system will not, generally speaking, function on another operating system unless it is first adapted, or ``ported," to the APIs of the other operating system.

6. In 1981, Microsoft released the first version of its Microsoft Disk Operating System, commonly known as “MS-DOS.” The system had a character-based user interface that required the user to type specific instructions at a command prompt in order to perform tasks such as launching applications and copying files. When the International Business Machines Corporation (“IBM”) selected MS-DOS for pre-installation on its first generation of PCs, Microsoft's product became the predominant operating system sold for Intel-compatible PCs.

7. In 1985, Microsoft began shipping a software package called Windows. The product included a graphical user interface, which enabled users to perform tasks by selecting icons and words on the screen using a mouse. Although originally just a user-interface, or “shell,” sitting on top of MS-DOS, Windows took on more operating- system functionality over time.

8. In 1995, Microsoft introduced a software package called Windows 95, which announced itself as the first operating system for Intel-compatible PCs that exhibited the same sort of integrated features as the Mac OS running PCs manufactured by Apple Computer, Inc. (“Apple”). Windows 95 enjoyed unprecedented popularity with consumers, and in June 1998, Microsoft released its successor, Windows 98.

9. Microsoft is the leading supplier of operating systems for PCs. The company transacts business in all fifty of the United States and in most countries around the world.

10. Microsoft licenses copies of its software programs directly to consumers. The largest part of its MS-DOS and Windows sales, however, consists of licensing the products to manufacturers of PCs (known as “original equipment manufacturers” or “OEMs”), such as the IBM PC Company and the Compaq Computer Corporation (“Compaq”). An OEM typically installs a copy of Windows onto one of its PCs before selling the package to a consumer under a single price.

17. Although certain Web browsers provided graphical user interfaces as far back as 1993, the first widely-popular graphical browser distributed for profit, called Navigator, was brought to market by the Netscape Communications Corporation ("Netscape") in December 1994. Microsoft introduced its browser, called Internet Explorer, in July 1995.

18. Currently there are no products, nor are there likely to be any in the near future, that a significant percentage of consumers worldwide could substitute for Intel-compatible PC operating systems without incurring substantial costs. Furthermore, no firm that does not currently market Intel-compatible PC operating systems could start doing so in a way that would, within a reasonably short period of time, present a significant percentage of consumers with a viable alternative to existing Intel-compatible PC operating systems. It follows that, if one firm controlled the licensing of all Intel-compatible PC operating systems worldwide, it could set the price of a license substantially above that which would be charged in a competitive market and leave the price there for a significant period of time without losing so many customers as to make the action unprofitable. Therefore, in determining the level of Microsoft's market power, the relevant market is the licensing of all Intel-compatible PC operating systems worldwide.

20. Since only Intel-compatible PC operating systems will work with Intel-compatible PCs, a consumer cannot opt for a non-Intel-compatible PC operating system without obtaining a non-Intel-compatible PC. Thus, for consumers who already own an Intel-compatible PC system, the cost of switching to a non-Intel compatible PC operating system includes the price of not only a new operating system, but also a new PC and new peripheral devices. It also includes the effort of learning to use the new system, the cost of acquiring a new set of compatible applications, and the work of replacing files and documents that were associated with the old applications. Very few consumers would incur these costs in response to the trivial increase in the price of an Intel-compatible PC system that would result from even a substantial increase in the price of an Intel-compatible PC operating system. For example, users of Intel-compatible PC operating systems would not switch in large numbers to the Mac OS in response to even a substantial, sustained increase in the price of an Intel-compatible PC operating system.

28. Operating systems are not the only software programs that expose APIs to application developers. Netscape's Web browser and Sun Microsystems, Inc.'s Java class libraries are examples of non-operating system software that do likewise. Such software is often called "middleware," because it relies on the interfaces provided by the underlying operating system while simultaneously exposing its own APIs to developers. Currently no middleware product exposes enough APIs to allow independent software vendors ("ISVs") profitably to write full-featured personal productivity applications that rely solely on those APIs.

29. Even if middleware deployed enough APIs to support full-featured applications, it would not function on a computer without an operating system to perform tasks such as managing hardware resources and controlling peripheral devices. But to the extent the array of applications relying solely on middleware comes to satisfy all of a user's needs, the user will not care whether there exists a large number of other applications that are directly compatible with the underlying operating system. Thus, the growth of middleware- based applications could lower the costs to users of choosing a non-Intel-compatible PC operating system like the Mac OS. It remains to be seen, though, whether there will ever be a sustained stream of full-featured applications written

solely to middleware APIs. In any event, it would take several years for middleware and the applications it supports to evolve from the status quo to a point at which the cost to the average consumer of choosing a non-Intel compatible PC operating system over an Intel-compatible one falls so low as to constrain the pricing of the latter systems.

30. Firms that do not currently produce Intel-compatible PC operating systems could do so. What is more, once a firm had written the necessary software code, it could produce millions of copies of its operating system at relatively low cost. The ability to meet a large demand is useless, however, if the demand for the product is small, and signs do not indicate large demand for a new Intel-compatible PC operating system. To the contrary, they indicate that the demand for a new Intel-compatible PC operating system would be severely constrained by an intractable “chicken-and-egg” problem: The overwhelming majority of consumers will only use a PC operating system for which there already exists a large and varied set of high-quality, full-featured applications, and for which it seems relatively certain that new types of applications and new versions of existing applications will continue to be marketed at pace with those written for other operating systems. Unfortunately for firms whose products do not fit that bill, the porting of applications from one operating system to another is a costly process. Consequently, software developers generally write applications first, and often exclusively, for the operating system that is already used by a dominant share of all PC users. Users do not want to invest in an operating system until it is clear that the system will support generations of applications that will meet their needs, and developers do not want to invest in writing or quickly porting applications for an operating system until it is clear that there will be a sizeable and stable market for it. What is more, consumers who already use one Intel-compatible PC operating system are even less likely than first-time buyers to choose a newcomer to the field, for switching to a new system would require these users to scrap the investment they have made in applications, training, and certain hardware.

31. The chicken-and-egg problem notwithstanding, a firm might reasonably expect to make a profit by introducing an Intel-compatible PC operating system designed to support a type of application that satisfies the special interests of a particular subset of users. For example, Be, Inc. (“Be”) markets an Intel-compatible PC operating system called BeOS that offers superior support for multimedia applications, and the operating system enjoys a certain amount of success with the segment of the consumer population that has a special interest in creating and playing multimedia content with a PC system. Still, while a niche operating system might turn a profit, the chicken-and-egg problem (hereinafter referred to as the “applications barrier to entry'”) would make it prohibitively expensive for a new Intel-compatible operating system to attract enough developers and consumers to become a viable alternative to a dominant incumbent in less than a few years.

32. To the extent that developers begin writing attractive applications that rely solely on servers or middleware instead of PC operating systems, the applications barrier to entry could erode. As the Court finds above, however, it remains to be seen whether server- or middleware-based development will flourish at all. Even if such development were already flourishing, it would still be several years before the applications barrier eroded enough to clear the way for the relatively rapid emergence of a viable alternative to incumbent Intel-compatible PC operating systems. It is highly unlikely, then, that a firm not already marketing an Intel-compatible PC

operating system could begin marketing one that would, in less than a few years, present a significant percentage of consumers with a viable alternative to incumbents.

33. Microsoft enjoys so much power in the market for Intel-compatible PC operating systems that if it wished to exercise this power solely in terms of price, it could charge a price for Windows substantially above that which could be charged in a competitive market. Moreover, it could do so for a significant period of time without losing an unacceptable amount of business to competitors. In other words, Microsoft enjoys monopoly power in the relevant market.

34. Viewed together, three main facts indicate that Microsoft enjoys monopoly power. First, Microsoft's share of the market for Intel-compatible PC operating systems is extremely large and stable. Second, Microsoft's dominant market share is protected by a high barrier to entry. Third, and largely as a result of that barrier, Microsoft's customers lack a commercially viable alternative to Windows.

35. Microsoft possesses a dominant, persistent, and increasing share of the worldwide market for Intel-compatible PC operating systems. Every year for the last decade, Microsoft's share of the market for Intel-compatible PC operating systems has stood above ninety percent. For the last couple of years, the figure has been at least ninety-five percent, and analysts project that the share will climb even higher over the next few years. Even if Apple's Mac OS were included in the relevant market, Microsoft's share would still stand well above eighty percent.

36. Microsoft's dominant market share is protected by the same barrier that helps define the market for Intel-compatible PC operating systems. As explained above, the applications barrier would prevent an aspiring entrant into the relevant market from drawing a significant number of customers away from a dominant incumbent even if the incumbent priced its products substantially above competitive levels for a significant period of time. Because Microsoft's market share is so dominant, the barrier has a similar effect within the market: It prevents Intel-compatible PC operating systems other than Windows from attracting significant consumer demand, and it would continue to do so even if Microsoft held its prices substantially above the competitive level.

37. Consumer interest in a PC operating system derives primarily from the ability of that system to run applications. The consumer wants an operating system that runs not only types of applications that he knows he will want to use, but also those types in which he might develop an interest later. Also, the consumer knows that if he chooses an operating system with enough demand to support multiple applications in each product category, he will be less likely to find himself straitened later by having to use an application whose features disappoint him. Finally, the average user knows that, generally speaking, applications improve through successive versions. He thus wants an operating system for which successive generations of his favorite applications will be released - promptly at that. The fact that a vastly larger number of applications are written for Windows than for other PC operating systems attracts consumers to Windows, because it reassures them that their interests will be met as long as they use Microsoft's product.

38. Software development is characterized by substantial economies of scale. The fixed costs of producing software, including applications, is very high. By contrast, marginal costs are very low. Moreover, the costs of developing software are “sunk” - once expended to develop software, resources so devoted cannot be used for another purpose. The result of economies of scale and sunk costs is that application developers seek to sell as many copies of their applications as possible. An application that is written for one PC operating system will operate on another PC operating system only if it is ported to that system, and porting applications is both time-consuming and expensive. Therefore, application developers tend to write first to the operating system with the most users - Windows. Developers might then port their applications to other operating systems, but only to the extent that the marginal added sales justify the cost of porting. In order to recover that cost, ISVs that do go to the effort of porting frequently set the price of ported applications considerably higher than that of the original versions written for Windows.

39. Consumer demand for Windows enjoys positive network effects. A positive network effect is a phenomenon by which the attractiveness of a product increases with the number of people using it. The fact that there is a multitude of people using Windows makes the product more attractive to consumers. The large installed base attracts corporate customers who want to use an operating system that new employees are already likely to know how to use, and it attracts academic consumers who want to use software that will allow them to share files easily with colleagues at other institutions. The main reason that demand for Windows experiences positive network effects, however, is that the size of Windows' installed base impels ISVs to write applications first and foremost to Windows, thereby ensuring a large body of applications from which consumers can choose. The large body of applications thus reinforces demand for Windows, augmenting Microsoft's dominant position and thereby perpetuating ISV incentives to write applications principally for Windows. This self-reinforcing cycle is often referred to as a “positive feedback loop.”

44. Microsoft continually releases “new and improved” versions of its PC operating system. Each time it does, Microsoft must convince ISVs to write applications that take advantage of new APIs, so that existing Windows users will have incentive to buy an upgrade. Since ISVs are usually still earning substantial revenue from applications written for the last version of Windows, Microsoft must convince them to write for the new version. Even if ISVs are slow to take advantage of the new APIs, though, no applications barrier stands in the way of consumers adopting the new system, for Microsoft ensures that successive versions of Windows retain the ability to run applications developed for earlier versions. In fact, since ISVs know that consumers do not feel locked into their old versions of Windows and that new versions have historically attracted substantial consumer demand, ISVs will generally write to new APIs as long as the interfaces enable attractive, innovative features. Microsoft supplements developers' incentives by extending various “seals of approval” - visible to consumers, investors, and industry analysts - to those ISVs that promptly develop new versions of their applications adapted to the newest version of Windows. In addition, Microsoft works closely with ISVs to help them adapt their applications to the newest version of the operating system - a process that is in any event far easier than porting an application from one vendor's PC operating system to another's. In sum, despite the substantial resources Microsoft expends inducing ISVs to develop applications for new versions of Windows, the company does not face any obstacles nearly as

imposing as the barrier to entry that vendors and would-be vendors of other PC operating systems must overcome.

59. The software industry in general is characterized by dynamic, vigorous competition. In many cases, one of the early entrants into a new software category quickly captures a lion's share of the sales, while other products in the category are either driven out altogether or relegated to niche positions. What eventually displaces the leader is often not competition from another product within the same software category, but rather a technological advance that renders the boundaries defining the category obsolete. These events, in which categories are redefined and leaders are superseded in the process, are spoken of as “inflection points.”

60. The exponential growth of the Internet represents an inflection point born of complementary technological advances in the computer and telecommunications industries. The rise of the Internet in turn has fueled the growth of server-based computing, middleware, and open-source software development. Working together, these nascent paradigms could oust the PC operating system from its position as the primary platform for applications development and the main interface between users and their computers. Microsoft recognizes that new paradigms could arise to depreciate the value of selling PC operating systems; however, the fact that these new paradigms already exist in embryonic or primitive form does not prevent Microsoft from enjoying monopoly power today. For while consumers might one day turn to network computers, or Linux, or a combination of middleware and some other operating system, as an alternative to Windows, the fact remains that they are not doing so today. Nor are consumers likely to do so in appreciable numbers any time in the next few years. Unless and until that day arrives, no significant percentage of consumers will be able to abandon Windows without incurring substantial costs. Microsoft can therefore set the price of Windows substantially higher than that which would be charged in a competitive market - or impose other burdens on consumers - without losing so much business as to make the action unprofitable. If Microsoft exerted its power solely to raise price, the day when users could turn away from Windows without incurring substantial costs would still be several years distant. Moreover, Microsoft could keep its prices high for a significant period of time and still lower them in time to meet the threat of a new paradigm. Alternatively, Microsoft could delay the arrival of a new paradigm on the scene by expending surplus monopoly power in ways other than the maintenance of high prices.

68. Middleware technologies, as previously noted, have the potential to weaken the applications barrier to entry. Microsoft was apprehensive that the APIs exposed by middleware technologies would attract so much developer interest, and would become so numerous and varied, that there would arise a substantial and growing number of full-featured applications that relied largely, or even wholly, on middleware APIs. The applications relying largely on middleware APIs would potentially be relatively easy to port from one operating system to another. The applications relying exclusively on middleware APIs would run, as written, on any operating system hosting the requisite middleware. So the more popular middleware became and the more APIs it exposed, the more the positive feedback loop that sustains the applications barrier to entry would dissipate. Microsoft was concerned with middleware as a category of software; each type of middleware contributed to the threat posed by the entire category. At the same time, Microsoft focused its antipathy on two incarnations of middleware that, working together, had the potential to weaken the applications barrier severely without the assistance of

any other middleware. These were Netscape's Web browser and Sun's implementation of the Java technologies.

69. Netscape Navigator possesses three key middleware attributes that endow it with the potential to diminish the applications barrier to entry. First, in contrast to non-Microsoft, Intel-compatible PC operating systems, which few users would want to use on the same PC systems that carry their copies of Windows, a browser can gain widespread use based on its value as a complement to Windows. Second, because Navigator exposes a set (albeit a limited one) of APIs, it can serve as a platform for other software used by consumers. A browser product is particularly well positioned to serve as a platform for network-centric applications that run in association with Web pages. Finally, Navigator has been ported to more than fifteen different operating systems. Thus, if a developer writes an application that relies solely on the APIs exposed by Navigator, that application will, without any porting, run on many different operating systems.

70. Adding to Navigator's potential to weaken the applications barrier to entry is the fact that the Internet has become both a major inducement for consumers to buy PCs for the first time and a major occupier of the time and attention of current PC users. For any firm looking to turn its browser product into an applications platform such to rival Windows, the intense consumer interest in all things Internet-related is a great boon.

71. Microsoft knew in the fall of 1994 that Netscape was developing versions of a Web browser to run on different operating systems. It did not yet know, however, that Netscape would employ Navigator to generate revenue directly, much less that the product would evolve in such a way as to threaten Microsoft. In fact, in late December 1994, Netscape's chairman and chief executive officer (“CEO”), Jim Clark, told a Microsoft executive that the focus of Netscape's business would be applications running on servers and that Netscape did not intend to succeed at Microsoft's expense.

72. As soon as Netscape released Navigator on December 15, 1994, the product began to enjoy dramatic acceptance by the public; shortly after its release, consumers were already using Navigator far more than any other browser product. This alarmed Microsoft, which feared that Navigator's enthusiastic reception could embolden Netscape to develop Navigator into an alternative platform for applications development. In late May 1995, Bill Gates, the chairman and CEO of Microsoft, sent a memorandum entitled “The Internet Tidal Wave” to Microsoft's executives describing Netscape as a “new competitor ‘born’ on the Internet.” He warned his colleagues within Microsoft that Netscape was “pursuing a multi-platform strategy where they move the key API into the client to commoditize the underlying operating system.” By the late spring of 1995, the executives responsible for setting Microsoft's corporate strategy were deeply concerned that Netscape was moving its business in a direction that could diminish the applications barrier to entry.

73. The term “Java” refers to four interlocking elements. First, there is a Java programming language with which developers can write applications. Second, there is a set of programs written in Java that expose APIs on which developers writing in Java can rely. These programs are called the “Java class libraries.” The third element is the Java compiler, which translates the

code written by the developer into Java “bytecode.” Finally, there are programs called “Java virtual machines,” or “JVMs,” which translate Java bytecode into instructions comprehensible to the underlying operating system. If the Java class libraries and a JVM are present on a PC system, the system is said to carry a “Java runtime environment.”

74. The inventors of Java at Sun Microsystems intended the technology to enable applications written in the Java language to run on a variety of platforms with minimal porting. A program written in Java and relying only on APIs exposed by the Java class libraries will run on any PC system containing a JVM that has itself been ported to the resident operating system. Therefore, Java developers need to port their applications only to the extent that those applications rely directly on the APIs exposed by a particular operating system. The more an application written in Java relies on APIs exposed by the Java class libraries, the less work its developer will need to do to port the application to different operating systems. The easier it is for developers to port their applications to different operating systems, the more applications will be written for operating systems other than Windows. To date, the Java class libraries do not expose enough APIs to support the development of full-featured applications that will run well on multiple operating systems without the need for porting; however, they do allow relatively simple, network-centric applications to be written cross-platform. It is Sun's ultimate ambition to expand the class libraries to such an extent that many full-featured, end-user-oriented applications will be written cross-platform. The closer Sun gets to this goal of “write once, run anywhere,” the more the applications barrier to entry will erode.

75. Sun announced in May 1995 that it had developed the Java programming language. Mid-level executives at Microsoft began to express concern about Sun's Java vision in the fall of that year, and by late spring of 1996, senior Microsoft executives were deeply worried about the potential of Sun's Java technologies to diminish the applications barrier to entry.

76. Sun's strategy could only succeed if a Java runtime environment that complied with Sun's standards found its way onto PC systems running Windows. Sun could not count on Microsoft to ship with Windows an implementation of the Java runtime environment that threatened the applications barrier to entry. Fortunately for Sun, Netscape agreed in May 1995 to include a copy of Sun's Java runtime environment with every copy of Navigator, and Navigator quickly became the principal vehicle by which Sun placed copies of its Java runtime environment on the PC systems of Windows users.

77. The combined efforts of Netscape and Sun threatened to hasten the demise of the applications barrier to entry, opening the way for non-Microsoft operating systems to emerge as acceptable substitutes for Windows. By stimulating the development of network-centric Java applications accessible to users through browser products, the collaboration of Netscape and Sun also heralded the day when vendors of information appliances and network computers could present users with viable alternatives to PCs themselves. Nevertheless, these middleware technologies have a long way to go before they might imperil the applications barrier to entry. Windows 98 exposes nearly ten thousand APIs, whereas the combined APIs of Navigator and the Java class libraries, together representing the greatest hope for proponents of middleware, total less than a thousand. Decision-makers at Microsoft are apprehensive of potential as well as present threats, though, and in 1995 the implications of the symbiosis between Navigator and

Sun's Java implementation were not lost on executives at Microsoft, who viewed Netscape's cooperation with Sun as a further reason to dread the increasing use of Navigator.

143. Decision-makers at Microsoft worried that simply developing its own attractive browser product, pricing it at zero, and promoting it vigorously would not divert enough browser usage from Navigator to neutralize it as a platform. They believed that a comparable browser product offered at no charge would still not be compelling enough to consumers to detract substantially from Navigator's existing share of browser usage. This belief was due, at least in part, to the fact that Navigator already enjoyed a very large installed base and had become nearly synonymous with the Web in the public's consciousness. If Microsoft was going to raise Internet Explorer's share of browser usage and lower Navigator's share, executives at Microsoft believed they needed to constrict Netscape's access to the distribution channels that led most efficiently to browser usage.

144. Very soon after it recognized the need to gain browser usage share at Navigator's expense, Microsoft identified pre-installation by OEMs and bundling with the proprietary client software of IAPs as the two distribution channels that lead most efficiently to browser usage. Two main reasons explain why these channels are so efficient. First, users must acquire a computer and connect to the Internet before they can browse the Web. Thus, the OEM and IAP channels lead directly to virtually every user of browsing software. Second, both OEMs and IAPs are able to place browsing software at the immediate disposal of a user without any effort on the part of the user. If an OEM pre-installs a browser onto its PCs and places an icon for that browser on the default screen, or “desktop,” of the operating system, purchasers of those PCs will be confronted with the icon as soon as the operating system finishes loading into random access memory (“RAM”). If an IAP bundles a browser with its own proprietary software, its subscribers will, by default, use the browser whenever they connect to the Web. In its internal decision-making, Microsoft has placed considerable reliance on studies showing that consumers tend strongly to use whatever browsing software is placed most readily at their disposal, and that once they have acquired, found, and used one browser product, most are reluctant - and indeed have little reason - to expend the effort to switch to another. Microsoft has also relied on studies showing that a very large majority of those who browse the Web obtain their browsing software with either their PCs or their IAP subscriptions.

145. Indeed, no other distribution channel for browsing software even approaches the efficiency of OEM pre-installation and IAP bundling. The primary reason is that the other channels require users to expend effort before they can start browsing. The traditional retail channel, for example, requires the consumer to make contact with a retailer, and retailers generally do not distribute products without charging a price for them. Naturally, once Microsoft and Netscape began offering browsing software for free, consumers for the most part lost all incentive to pay for it.

148. Knowing that OEMs and IAPs represented the most efficient distribution channels of browsing software, Microsoft sought to ensure that, to as great an extent as possible, OEMs and IAPs bundled and promoted Internet Explorer to the exclusion of Navigator.

158. Microsoft did manage to bundle Internet Explorer 1.0 with the first version of Windows 95 licensed to OEMs in July 1995. It also included a term in its OEM licenses that prohibited the OEMs from modifying or deleting any part of Windows 95, including Internet Explorer, prior to shipment. The OEMs accepted this restriction despite their interest in meeting consumer demand for PC operating systems without Internet Explorer. After all, Microsoft made the restriction a non-negotiable term in its Windows 95 license, and the OEMs felt they had no commercially viable alternative to pre-installing Windows 95 on their PCs. Apart from a few months in the fall of 1997, when Microsoft provided OEMs with Internet Explorer 4.0 on a separate disk from Windows 95 and permitted them to ship the latter without the former, Microsoft has never allowed OEMs to ship Windows 95 to consumers without Internet Explorer. This policy has guaranteed the presence of Internet Explorer on every new Windows PC system.

159. Microsoft knew that the inability to remove Internet Explorer made OEMs less disposed to pre-install Navigator onto Windows 95. OEMs bear essentially all of the consumer support costs for the Windows PC systems they sell. These include the cost of handling consumer complaints and questions generated by Microsoft's software. Pre-installing more than one product in a given category, such as word processors or browsers, onto its PC systems can significantly increase an OEM's support costs, for the redundancy can lead to confusion among novice users. In addition, pre-installing a second product in a given software category can increase an OEM's product testing costs. Finally, many OEMs see pre-installing a second application in a given software category as a questionable use of the scarce and valuable space on a PC's hard drive.

160. Microsoft's executives believed that the incentives that its contractual restrictions placed on OEMs would not be sufficient in themselves to reverse the direction of Navigator's usage share. Consequently, in late 1995 or early 1996, Microsoft set out to bind Internet Explorer more tightly to Windows 95 as a technical matter. The intent was to make it more difficult for anyone, including systems administrators and users, to remove Internet Explorer from Windows 95 and to simultaneously complicate the experience of using Navigator with Windows 95. As Brad Chase wrote to his superiors near the end of 1995, “We will bind the shell to the Internet Explorer, so that running any other browser is a jolting experience.”

161. Microsoft bound Internet Explorer to Windows 95 by placing code specific to Web browsing in the same files as code that provided operating system functions. Starting with the release of Internet Explorer 3.0 and “OEM Service Release 2.0” (“OSR 2”) of Windows 95 in August 1996, Microsoft offered only a version of Windows 95 in which browsing-specific code shared files with code upon which non-browsing features of the operating system relied.

164. Starting with Windows 95 OSR 2, Microsoft placed many of the routines that are used by Internet Explorer, including browsing-specific routines, into the same files that support the 32-bit Windows APIs. Microsoft's primary motivation for this action was to ensure that the deletion of any file containing browsing-specific routines would also delete vital operating system routines and thus cripple Windows 95. Although some of the code that provided Web browsing could still be removed, without disabling the operating system, by entering individual files and selectively deleting routines used only for Web browsing, licensees of Microsoft software were, and are, contractually prohibited from reverse engineering, decompiling, or disassembling any software

files. Even if this were not so, it is prohibitively difficult for anyone who does not have access to the original, human-readable source code to change the placement of routines into files, or otherwise to alter the internal configuration of software files, while still preserving the software's overall functionality.

203. If OEMs removed the most visible means of invoking Internet Explorer, and pre-installed Navigator with facile methods of access, Microsoft's purpose in forcing OEMs to take Internet Explorer - capturing browser usage share from Netscape - would be subverted. The same would be true if OEMs simply configured their machines to promote Navigator before Windows had a chance to promote Internet Explorer. Decision-makers at Microsoft believed that as Internet Explorer caught up with Navigator in quality, OEMs would ultimately conclude that the costs of pre-installing and promoting Navigator, and removing easy access to Internet Explorer, outweighed the benefits. Still, those decision-makers did not believe that Microsoft could afford to wait for the several large OEMs that represented virtually all Windows PCs shipped to come to this desired conclusion on their own. Therefore, in order to bring the behavior of OEMs into line with its strategic goals quickly, Microsoft threatened to terminate the Windows license of any OEM that removed Microsoft's chosen icons and program entries from the Windows desktop or the “Start” menu. It threatened similar punishment for OEMs that added programs that promoted third-party software to the Windows “boot” sequence. These inhibitions soured Microsoft's relations with OEMs and stymied innovation that might have made Windows PC systems more satisfying to users. Microsoft would not have paid this price had it not been convinced that its actions were necessary to ostracize Navigator from the vital OEM distribution channel.

213. In an effort to thwart the practice of OEM customization, Microsoft began, in the spring of 1996, to force OEMs to accept a series of restrictions on their ability to reconfigure the Windows 95 desktop and boot sequence. There were five such restrictions, which were manifested either as amendments to existing Windows 95 licenses or as terms in new Windows 98 licenses. First, Microsoft formalized the prohibition against removing any icons, folders, or “Start” menu entries that Microsoft itself had placed on the Windows desktop. Second, Microsoft prohibited OEMs from modifying the initial Windows boot sequence. Third, Microsoft prohibited OEMs from installing programs, including alternatives to the Windows desktop user interface, which would launch automatically upon completion of the initial Windows boot sequence. Fourth, Microsoft prohibited OEMs from adding icons or folders to the Windows desktop that were not similar in size and shape to icons supplied by Microsoft. Finally, when Microsoft later released the Active Desktop as part of Internet Explorer 4.0, it added the restriction that OEMs were not to use that feature to display third-party brands.

214. The several OEMs that in the aggregate represented over ninety percent of Intel-compatible PC sales believed that the new restrictions would make their PC systems more difficult and more confusing to use, and thus less acceptable to consumers. They also anticipated that the restrictions would increase product returns and support costs and generally lower the value of their machines. Those OEMs that had already spent millions of dollars developing and implementing tutorial and registration programs and/or automatically-loading graphical interfaces in the Windows boot sequence lamented that their investment would, as a result of Microsoft's policy, be largely wasted. Gateway, Hewlett-Packard, and IBM communicated their

opposition forcefully and urged Microsoft to lift the restrictions. Emblematic of the reaction among large OEMs was a letter that the manager of research and development at Hewlett-Packard sent to Microsoft in March 1997. He wrote:

> “Microsoft's mandated removal of all OEM boot-sequence and auto-start programs for OEM licensed systems has resulted in significant and costly problems for the HP-Pavilion line of retail PC's.
>
> Our data (as of 3/10/97) shows a 10% increase in W[indows]95 calls as a % of our total customer support calls….
>
> Our registration rate has also dropped from the mid-80% range to the low 60% range.
>
> There is also subjective data from several channel partners that our system return rate has increased from the lowest of any OEM (even lower than Apple) to a level comparable to the other Microsoft OEM PC vendors. This is a major concern in that we are taking a step backward in meeting customer satisfaction needs. These three pieces of data confirm that we have been damaged by the edicts that [ ] Microsoft issued last fall.
>
> From the consumer perspective, we are hurting our industry and our customers. PC's can be frightening and quirky pieces of technology into which they invest a large sum of their money. It is vitally important that the PC suppliers dramatically improve the consumer buying experience, out of box experience as well as the longer term product usability and reliability. The channel feedback as well as our own data shows that we are going in the wrong direction. This causes consumer dissatisfaction in complex telephone support process, needless in-home repair visits and ultimately in product returns. Many times the cause is user misunderstanding of a product that presents too much complexity to the common user….
>
> Our Customers hold HP accountable for their dissatisfaction with our products. We bear [ ] the cost of returns of our products. We are responsible for the cost of technical support of our customers, including the 33% of calls we get related to the lack of quality or confusion generated by your product. And finally we are responsible for our success or failure in the retail PC market.
>
> We must have more ability to decide how our system is presented to our end users.
>
> If we had a choice of another supplier, based on your actions in this area, I assure you [that you] would not be our supplier of choice.
>
> I strongly urge you to have your executives review these decisions and to change this unacceptable policy.”

227. To the extent Microsoft is apprehensive that OEMs might, absent restrictions, change the set of APIs exposed by the software on their PCs, the concern is not that OEMs would modify the Windows API set. Rather, the worry is that OEMs would pre-install, on top of Windows, other software exposing additional APIs not controlled by Microsoft. In the case of alternate user interfaces, Microsoft is fearful that, if these programs loaded automatically the first time users

turned on their PCs, the programs would attract so much usage that developers would be encouraged to take advantage of any APIs that the programs exposed. Indeed, one user interface in particular that OEMs could configure to load automatically and obscure the Windows desktop – Navigator- exposes a substantial number of APIs. Therefore, Microsoft's real concern has not been that OEM modifications would fragment the Windows platform to the detriment of developers and consumers. What has motivated Microsoft's prohibition against automatically loading shells is rather the fear - once again - that OEMs would preinstall and give prominent placement to middleware that could weaken the applications barrier to entry.

239. Microsoft has largely succeeded in exiling Navigator from the crucial OEM distribution channel. Even though a few OEMs continue to offer Navigator on some of their PCs, Microsoft has caused the number of OEMs offering Navigator, and the number of PCs on which they offer it, to decline dramatically. Before 1996, Navigator enjoyed a substantial and growing presence on the desktop of new PCs. Over the next two years, however, Microsoft's actions forced the number of copies of Navigator distributed through the OEM channel down to an exiguous fraction of what it had been. By January 1998, Kempin could report to his superiors at Microsoft that, of the sixty OEM sub-channels (fifteen major OEMs each offering corporate desktop, consumer/small business, notebook, and workstation PCs), Navigator was being shipped through only four. Furthermore, most of the PCs shipped with Navigator featured the product in a manner much less likely to lead to usage than if its icon appeared on the desktop. For example, Sony only featured Navigator in a folder rather than on the desktop, and Gateway only shipped Navigator on a separate CD-ROM rather than pre-installed on the hard drive. By the beginning of January 1999, Navigator was present on the desktop of only a tiny percentage of the PCs that OEMs were shipping.

339. In dozens of “First Wave” agreements signed between the fall of 1997 and the spring of 1998, Microsoft has promised to give preferential support, in the form of early Windows 98 and Windows NT betas, other technical information, and the right to use certain Microsoft seals of approval, to important ISVs that agree to certain conditions. One of these conditions is that the ISVs use Internet Explorer as the default browsing software for any software they develop with a hypertext-based user interface. Another condition is that the ISVs use Microsoft's “HTML Help,” which is accessible only with Internet Explorer, to implement their applications' help systems.

340. By exchanging its vital support for the agreement of leading ISVs to make Internet Explorer the default browsing software on which their products rely, Microsoft has ensured that many of the most popular Web-centric applications will rely on browsing technologies found only in Windows and has increased the likelihood that the millions of consumers using these products will use Internet Explorer rather than Navigator. Microsoft's relations with ISVs thus represent another area in which it has applied its monopoly power to the task of protecting the applications barrier to entry.

394. In a further effort intended to increase the incompatibility between Java applications written for its Windows JVM and other Windows JVMs, and to increase the difficulty of porting Java applications from the Windows environment to other platforms, Microsoft designed its Java developer tools to encourage developers to write their Java applications using certain

“keywords” and “compiler directives” that could only be executed properly by Microsoft's version of the Java runtime environment for Windows. Microsoft encouraged developers to use these extensions by shipping its developer tools with the extensions enabled by default and by failing to warn developers that their use would result in applications that might not run properly with any runtime environment other than Microsoft's and that would be difficult, and perhaps impossible, to port to JVMs running on other platforms. This action comported with the suggestion that Microsoft's Thomas Reardon made to his colleagues in November 1996: “[W]e should just quietly grow j++ [Microsoft's developer tools] share and assume that people will take more advantage of our classes without ever realizing they are building win32-only java apps.” Microsoft refused to alter its developer tools until November 1998, when a court ordered it to disable its keywords and compiler directives by default and to warn developers that using Microsoft's Java extensions would likely cause incompatibilities with non-Microsoft runtime environments.